

LIETUVOS TEISĖS UNIVERSITETAS

Dalė Dzemydienė, Ramutė Naujikienė

**INFORMACINĖS SISTEMOS.
DUOMENŲ STRUKTŪROS
IR VALDYMAS**

Vadovėlis

Vilnius 2004

UDK 004.42(075.8)
Dz-08

2003 m. gruodžio 19 d. Nr. A-179
Aukštųjų mokyklų bendrųjų vadovėlių leidybos komisijos
rekomenduota

R e c e n z a v o :

Vilniaus Gedimino technikos universiteto Fundamentinių mokslų fakulteto Informacinių sistemų katedros vedėjas prof. habil. dr. **Petras Gailutis Adomėnas**;

Vilniaus pedagoginio universiteto Matematikos ir informatikos fakulteto Informatikos katedros profesorius habil. dr. **Vydūnas Šaltenis**

Matematikos ir informatikos instituto Senato 2003 m. balandžio 16 d. posėdžio sprendimu (protokolas Nr. 2) vadovėlis rekomenduotas spausdinti

Lietuvos teisės universiteto Valstybinio valdymo fakulteto Teisinės informatikos katedros 2003 m. kovo 31 d. posėdžio sprendimu (protokolas Nr. 1 TIK-5) vadovėlis rekomenduotas spausdinti

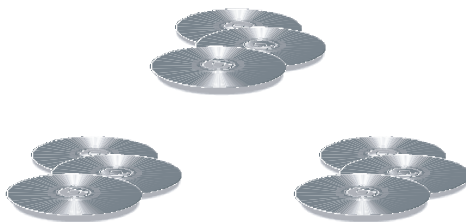
Lietuvos teisės universiteto vadovėlių, monografijų, mokslinių, mokomųjų, metodinių bei kitų leidinių aprobavimo spaudai komisija 2003 m. balandžio 17 d. posėdyje (protokolas Nr. 2L-4) vadovėlį patvirtino spausdinti

TURINYS

IŽANGA.....	6
1 skyrius. INFORMACIJOS VAIDMUO VALDYMO PROCESUOSE.....	9
1.1. Informacijos išteklių turinys ir formos.....	10
1.2. Duomenys, informacija, žinios.....	12
1.3. Informacijos vaidmuo skirtinguose valdymo lygmenyse.....	14
1.4. Duomenų struktūros taikomiosiose ir sisteminėse programose. 18	
1.4.1. Paprastieji, struktūriniai ir abstraktieji duomenų tipai.....	20
1.4.2. Dinaminės duomenų struktūros.....	21
1.4.3. Rodyklės.....	23
1.4.4. Dinaminis sąrašas.....	24
1.4.5. Dėklas.....	25
1.4.6. Eilė.....	25
1.4.7. Dvikryptis sąrašas.....	27
1.4.8. Medžio tipo struktūros.....	29
2 skyrius. INFORMACINĖS SISTEMOS.....	31
2.1. Informacinių sistemų samprata.....	32
2.2. Informacinės sistemos gyvavimo ciklas.....	33
2.3. Informacinės sistemos analizė.....	35
2.4. Informacinės sistemos koncepcijos aprašymas.....	36
2.5. Objektinė informacinių sistemų projektavimo metodika.....	38
2.5.1. Objektų klasės.....	39
2.5.2. Klasių dalijimas ir skirstymas.....	39
2.5.3. Objektų klasių ryšių tipai.....	40
2.6. Administravimo informacinės sistemos projektavimas.....	45
2.6.1. Konferencijos organizavimo ir administravimo informacinė sistema.....	45
2.6.2. Tikslų nustatymas.....	46
2.6.3. Koncepcinis projektavimas.....	47
2.7. Dalykinės srities duomenų srautų modeliavimas.....	49
2.7.1. Žodinis dalykinės srities aprašymas.....	51
2.7.2. Pranešimų apie nusikaltimus tyrimo duomenų srautų modelis.....	52
2.8. Išskirstytosios informacinės sistemos.....	54
2.9. Informacinių sistemų kontrolė ir apsauga.....	57
3 skyrius. DUOMENŲ BAZĖS IR JŲ VALDYMO SISTEMOS.....	60
3.1. Duomenų bazės samprata.....	61
3.2. Klasikiniai duomenų bazių modeliai.....	62
3.2.1. Reliacinės duomenų bazės.....	62
3.2.2. Hierarchinis duomenų modelis.....	65
3.2.3. Tinklinis duomenų modelis.....	66

3.3. Reliacinių duomenų bazių projektavimas.....	67
3.3.1. Duomenų bazės rakto samprata.....	69
3.3.2. Duomenų bazės struktūros normalizacija.....	70
3.3.3. Pirma normalinė forma.....	70
3.3.4. Antra normalinė forma.....	73
3.3.5. Trečia normalinė forma.....	75
3.3.6. Ketvirta normalinė forma.....	76
3.4. Duomenų bazių valdymo sistemos.....	77
3.5. Pagrindiniai duomenų bazės kūrimo ir redagavimo etapai.....	79
3.6. Duomenų paieškos priemonės.....	81
3.7. Duomenų bazių praktiniai darbai ir metodiniai nurodymai.....	83
3.7.1. Konferencijos organizavimo duomenų bazės kūrimas....	83
3.7.2. Lentelių ryšių nustatymas duomenų bazėje.....	94
3.7.3. Duomenų įvedimo formos ir jų projektavimas.....	101
3.7.4. Duomenų paieška, rūšiavimas ir filtravimas duomenų bazėje.....	108
3.7.5. Dialoginių užklausų atlikimo būdai.....	118
3.7.6. Suvestinių duomenų užklausų atlikimo būdai.....	124
3.7.7. Kelias lenteles siejančių užklausų atlikimo būdai.....	128
3.7.8. Kryžminių suvestinių duomenų užklausa.....	131
3.7.9. Lentelių kūrimo, papildymo, įrašų šalinimo, duomenų keitimo užklausa.....	137
3.7.10. Ataskaitos duomenų bazėje.....	143
3.8. Informacijos pateikimo praktiniai darbai.....	151
3.8.1. Serijinių laiškų rašymas naudojant duomenų bazę.....	151
3.8.2. Interneto svetainės kūrimas programa <i>FrontPage</i>	161
4 skyrius. DUOMENŲ TVARKYMAS SKAIČIUOKLĖSE.....	172
4.1. Šiuolaikinės elektroninių lentelių tvarkymo galimybės.....	173
4.2. Skaičiuoklių darbo aplinka.....	175
4.2.1. Komandų ir kontekstinis meniu.....	177
4.2.2. Ląstelių koordinačių tipai.....	178
4.2.3. Įrankių juostos.....	179
4.3. Skaičiuoklių duomenų tipai ir formatai.....	179
4.4. Uždavinių sprendimo tvarka skaičiuoklėse.....	183
4.5. Darbo knygos sandara ir jos kūrimas skaičiuoklėse.....	183
4.5.1. Skaičiuoklių darbo knygos kūrimas.....	183
4.5.2. Ląstelės turinio pildymas ir redagavimas.....	185
4.5.3. Formulės įrašymas į ląstelę.....	187
4.5.4. Ląstelių masyvas.....	188
4.5.5. Elektroninės lentelės redagavimas ir formatavimas.....	190
4.5.6. Specialūs duomenų kopijavimo būdai.....	192
4.5.7. Darbo pabaigos komandos.....	194
4.5.8. Lentelių rinkinio puslapių formavimas.....	195
4.6. Formulų sudarymas skaičiuoklėse.....	198

4.7. Funkcijų panaudojimas.....	200
4.7.1. Matematinės ir trigonometrinės funkcijos.....	202
4.7.2. Statistinės funkcijos.....	206
4.7.3. Finansinės funkcijos.....	207
4.7.4. Laiko apskaitos funkcijos.....	209
4.7.5. Informacinės funkcijos.....	212
4.7.6. Loginės funkcijos.....	212
4.7.7. Paieškos ir koordinacių funkcijos.....	214
4.8. Duomenų bazių ir sąrašo analizės komandos.....	214
4.8.1. Duomenų rikiavimas – patogi duomenų organizavimo priemonė.....	216
4.8.2. Duomenų filtravimas.....	217
4.9. Klaidų reikšmės skaičiuoklių programoje.....	219
4.10. Grafinis duomenų pateikimas ir vizualizavimas.....	220
4.11. Skaičiuoklės darbo knygos spausdinimo būdai.....	223
4.12. Skaičiuoklių praktiniai darbai ir metodiniai nurodymai.....	227
4.12.1. Duomenų analizė skaičiuoklėse.....	227
4.12.2. Duomenų vizualizavimas. Diagramų sudarymas.....	235
4.12.3. Pajamų planavimas ir analizė.....	241
4.12.4. Atlyginimų apskaitos darbo lapo sudarymas.....	249
4.12.5. Pajamų mokesčio apskaita skaičiuojant atlyginimą.....	255
LITERATŪRA.....	261



IŽANGA

Sparčiai plečiantis informacinių sistemų kūrimo galimybės keičiasi daugelio žmogaus veiklos sričių darbo aplinka. Technologijų plėtra turi įtakos programinės įrangos kūrimo procesams, naujų metodų paieškai, skirtai informacijai ir žinioms kaupti, skleisti, valdyti.

Informacijos supratimas siejamas su veiklos kompiuterizavimo procesais, naujomis technikos rūšimis, informacijos apdorojimo, saugojimo ir perdavimo technologijomis. Informacija ir žinios tampa ekonominiu ištekliumi, jos yra labiausiai kvalifikuotos ir kūrybiškiausios intelektualiosios žmogaus veiklos produktas.

Informacija yra reikšminga šiuolaikiniame gyvenime, šiam procesui turi įtakos keletas veiksnių. Daugelyje dalykinių sričių didėja veiklos procesų įvairovė, spartėja kompiuterinių priemonių raida ir taikymas, įgalinantys supaprastinti šių procesų valdymą. Informacija tampa vis labiau prieinama ir efektyviau perimama. Informacinės visuomenės raidos procesai pereina į naują etapą, vadinamą žinių visuomenės kūrimo etapu, kuriame nemažiau svarbiu ištekliumi tampa žmogaus žinojimo veiksniai, praktika ir įgūdžiai bei jų išsaugojimo skaitmeninėje erdvėje metodai ir galimybės. Žinoma, pirminiai faktai, stebimų reiškinių charakteristikos, įvardijami kaip duomenys apie mūsų pasaulį, sudaro informacijos turinio pagrindą. Be tikslių duomenų negausime ir apibendrinančių rezultatų, negalėsime priimti reikiamų sprendimų. Todėl šioje knygoje daugiausia dėmesio bus skiriama duomenų struktūroms ir jų valdymo galimybėms, tinkamų programinių priemonių parinkimui.

Duomenys kompiuterinėse sistemose turi savo struktūras, valdomas programinėmis priemonėmis. Realus pasaulis kompiuteryje gali būti vaizduojamas pasitelkiant modeliavimo priemones. Todėl svarbu mokėti įvertinti ir pateikti šiuos modelius, sugebėti tinkamai pasirinkti modeliavimo metodus, grafinės notacijos būdus, padedančius išreikšti taikomosios srities ypatumus kompiuterinėse sistemose.

A autorių indėlis rašant vadovėlį yra toks: D. Dzemydienė parašė įžangą, 1, 2, 3.1–3.6, 4.12 skyrius. R. Naujikienė parašė 3.7–3.8, 4.1–4.11 skyrius. Knygoje įvertinta autorių ilgamečio darbo su studentais bei sudėtingų informacinių sistemų kūrimo patirtis. D. Dzemydienė atstovauja Lietuvos teisės universitetui ir Matematikos ir informatikos institutui, kuriame įgyta mokslinio ir pedagoginio darbo patirtis leido išsamiau pažvelgti į informacinių sistemų kūrimo procesą ir susisteminti dėstomą teorinę medžiagą. R. Naujikienė atstovauja Lietuvos teisės universitetui ir turi didelę darbo kompiuteriu mokymo patirtį, kuri leido parinkti originalias praktines užduotis ir metodinius nurodymus.

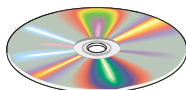
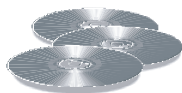
Esant dabartinei situacijai iš būsimojo specialisto pirmiausia bus reikalaujama efektyvaus darbo su naujomis informacinėmis technologijomis. Būsimiesiems valdymo specialistams svarbu suprasti organizacijų aplinką ir susipažinti su pagrindinėmis duomenų išraiškos priemonėmis, modeliais bei valdymo ypatumais, užtikrinančiais efektyvesnius darbo metodus. Reikšmingą vaidmenį šiame kontekste vaidina duomenų bazių valdymo sistemos. Jų teikiamos duomenų apdorojimo priemonės suteikta galimybę lengviau tvarkyti ir valdyti duomenis, juos išsaugoti, rūšiuoti, vykdyti paiešką, skleisti informaciją kompiuterių tinkluose. Savarankiški įgūdžiai, įgyti atliekant praktines užduotis pagal šiame vadovėlyje pateiktus metodinius nurodymus, skatins taikyti modernias kompiuterines technologijas ir ieškoti naujų galimybių susipažinti su šiuolaikinėmis informacinėmis sistemomis bei technologijomis.

Programinės įrangos kūrėjai, siūlantys savo produktus ir sistemas įstaigų darbuotojams, yra korporacijos *Microsoft*, *Borland*, *Lotus*, *IBM*, *Corel Systems* ir kt. Kuriamos atvirojo kodo sistemos, veikiančios *Linux* ar *Unix* operacinėje sistemoje. Šiuo metu langų valdymo operacinės sistemos autoriai siūlo šiai aplinkai sukurtą paketų rinkinį *Microsoft Office*, kuriame yra viena geriausių tekstų tvarkymo sistemų – *Word*, grafinis skaidrių rengimo paketas *PowerPoint*, elektroninių lentelių-skaičiuoklių valdymo programinė įranga *Excel*, duomenų bazių valdymo sistema *Access*, elektroninio pašto programa, interneto svetainių kūrimo priemonės *FrontPage* ir kt. Populiari šiuo metu naudojama skaičiuoklė *Microsoft Excel*. Pasaulyje, be šios programinės įrangos, dar naudojamos skaičiuoklės: *Lotus 1-2-3* (ją sukūrė *Lotus Development*, 1995 m. tapusi įmonės

IBM padaliniu) ir *Quattro Pro* (ją 1994 metais sukūrė įmonė *Borland International*). *Quattro Pro* skaičiuoklių valdymo sistema vėliau įėjo į *Novel* firmos *Perfect Office* programų rinkinį. Savo galimybėmis *Microsoft Excel* šiuo metu lenkia kitas minėtas sistemas. Ši programa turi daugiau kaip 300 funkcijų. Tai programa, su kuria patogiu ne tik skaičiuoti, bet ir įvairiapusiškai analizuoti duomenis, programuoti, atlikti statistinius tyrimus.

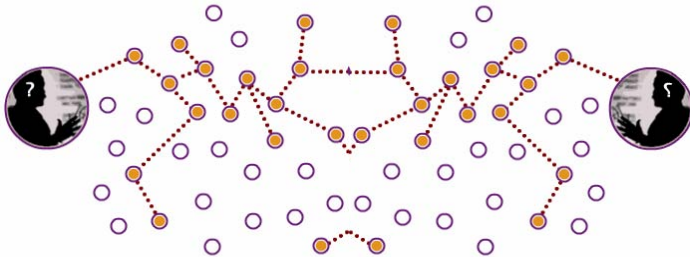
Vadovėlyje supažindinama su skaičiuoklėmis *Excel*, duomenų bazių valdymo sistema *Access*, interneto svetainių kūrimo priemonėmis. Autorių nuomone, ši knyga bus naudinga visiems norintiems susipažinti su šiuolaikinėmis informacinių sistemų kūrimo galimybėmis, išsiugdyti gebėjimą dirbti su duomenų valdymo struktūromis ir informacinėmis sistemomis.

Autorės dėkoja Lietuvos teisės universiteto Valstybinio valdymo fakulteto Teisinės informatikos katedros vedėjui profesoriui R. A. Petrauskui ir jos darbuotojams už patarimus, suteiktą programinę įrangą, kurios priemonėmis dėstytojai skatinami mokyti studentus naudotis šiuolaikinėmis informacinėmis technologijomis. Dėkojame Matematikos ir informatikos instituto Programų sistemų inžinerijos skyriaus darbuotojams ir vadovui profesoriui A. Čaplinskiui, kurių patarimai ir diskusijos padėjo tiksliau susisteminti dėstomą teorinę medžiagą. Dėkojame recenzentams profesoriui P. G. Adomėnui ir profesoriui V. Šalteniui už pateiktas pastabas. Nuoširdžiai dėkojame visiems prisidėjusiems prie šio darbo.



1 skyrius

INFORMACIJOS VAIDMUO VALDYMO PROCESUOSE



- 1.1. Informacijos išteklių turinys ir formos
- 1.2. Duomenys, informacija, žinios
- 1.3. Informacijos vaidmuo skirtinguose valdymo lygmenyse
- 1.4. Duomenų struktūros taikomosiose ir sisteminėse programose

1.1. INFORMACIJOS IŠTEKLIŲ TURINYS IR FORMOS

Informacinės sistemos kuriamos bei naudojamos daugelyje žmogaus veiklos sričių. Jos leidžia naudotis tikslia ir laiku gaunama informacija, yra skirtos palengvinti valdymo sprendimų priėmimą ir efektyviai atlikti planavimo, valdymo ir administravimo, koordinavimo ir kontrolės funkcijas. Tinkamai sukurtos informacinės valdymo sistemos teikia žinių apie organizacijos veiklos praeitį, dabartį, numatomą ateitį ir organizacijai svarbius vidaus bei išorės įvykius. Informacinės valdymo sistemos padeda analizuoti ir vizualizuoti informaciją, palengvina įvairaus lygio vadovų ir administratorių darbą.

Informacinėse valdymo sistemose esantys duomenys atspindi organizacijos, įmonės, įstaigos veiklos rezultatus ir aplinkos veiksnius, darančius įtaką šiai veiklai. Informacinėse sistemose informacija apie organizacijai ir aplinkai svarbius asmenis, veiklą, daiktus ir procesus turėtų būti kaupiama ir pateikiama žmogui patogia forma.

Informacijos samprata siejama su informacinės veiklos kompiuterizavimu, naujomis technikos rūšimis, šiuolaikinėmis informacijos apdorojimo, saugojimo ir perdavimo technologijomis [20; 25]. Kaupiama informacija ir žinios padeda taupyti visuomeninio, materialiojo, intelektinio darbo išteklius. Informacija atlieka strateginį vaidmenį šiuolaikinėje veikloje, sprendimų priėmimo procesuose.

Informacinės visuomenės raidos procesams pereinant į naują etapą, vadinamą žinių visuomenės kūrimo etapu, svarbiais ištekliais tampa informacija, žmogaus žinojimo veiksniai, praktika ir įgūdžiai bei jų išsaugojimas skaitmeninėje erdvėje. Atsižvelgiant į šiuolaikinio visuomenės išsivystymo lygį ypač daug dėmesio skiriama informacinių ir komunikacinių technologijų potencialui plėtoti [9; 14; 23]. Numatoma daugybė su šiomis technologijomis susijusių ekonominių, socialinių politinių veiklos krypčių. Su informacinių technologijų plėtros procesais susijusi Europos Sąjungos užimtumo politika, kuriami elektroninės Europos veiksmų planai. Informacinių, komunika-

cinių technologijų taikymas lemia ir naujų darbo formų atsiradimą, skatina sudaryti visų gyventojų užimtumo žinių visuomenėje sąlygas, pavyzdžiui, visiems prieinamą nuotolinį mokymąsi, visuomenės skaitmeninį raštingumą, viešųjų paslaugų modernizaciją [15; 19; 25].






Atsiranda naujos vadovavimo formos ir paslaugų rūšys: elektroninės vyriausybės kūrimas ir jos informacinių išteklių vadyba, elektroninė komercija, nuotolinis bendradarbiavimas [13; 20].

Šiuolaikinių informacinių ir komunikacinių technologijų plėtojimasis sudaro palankias sąlygas naudotis internetu ir kitomis mobiliojo ryšio priemonėmis. Atsiranda daugybė paslaugų, teikiamų per internetą, rūšių: elektroninė komercija, diskusijų svetainės, mokesčių mokėjimas, virtualios įstaigos ir nuotolinio projektavimo laboratorijos, duomenų bazės, dirbančios tiesioginės kreipties (*On-Line*) režimu.



Efektyvus informacinių ir komunikacinių technologijų taikymas skatina atsirasti naujas darbo organizavimo formas. Šie veiksniai daro pozityvią įtaką darbo kokybei: didina atsakomybę, darbo produktyvumą, padeda geriau valdyti informacijos srautus. Visa tai leidžia mėgautis darbu, padaryti jį efektyvesnį bei pertvarkyti darbą organizacijoje bei už jos ribų. Tokie pokyčiai vyksta daugelyje taikomųjų sričių: viešojo administravimo ir valdymo, medicinos diagnostikos, automatinio projektavimo, nuotolinio mokymo ir kitose srityse.

Informacinė sistema suprantama kaip struktūrizuotas procesų ir procedūrų rinkinys, kuriame duomenys yra kaupiami, organizuojami ir pateikiami vartotojui. Informacinė sistema – tai darbo praktikos, informacijos, žmonių ir informacinių technologijų, skirtų organizacijos tikslams siekti, kompleksas. Informacines sistemas sudaro programinė, techninė įranga, žmonės, duomenys.

Pagal turinį informacijos ištekliai skirstomi į:

-  Valdymo;
-  Teisinius;
-  Finansinius-ekonominius;
-  Mokslinius;
-  Reklaminius ir pan.

Pagrindiniai informacijos išteklių tipai:

-  Informacijos šaltinis;
-  Informacijos paslauga, kaip naudinga veikla;

- ✚ Informacijos produktas, kaip naudingas produktas, tenkinantis vartotojų poreikius;
- ✚ Informacinė sistema.

Pagal nuosavybės formą informacijos ištekliai gali būti skirstomi į:

- ✚ Valstybinius;
- ✚ Nevalstybinius;
- ✚ Mišrių nuosavybės formų.

Informacinių sistemų kokybę nusako:

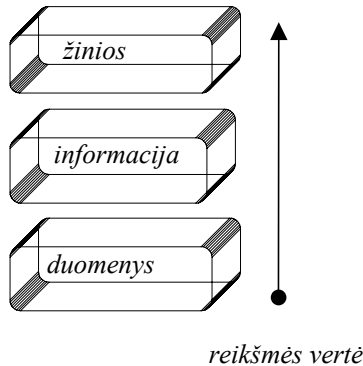
- ✚ Informacijos šaltinio patikimumas;
- ✚ Tikslumas;
- ✚ Operatyvumas;
- ✚ Paieškos kalbų paprastumas ir aiškumas;
- ✚ Minimalios paieškos laiko sąnaudos;
- ✚ Informacinės paieškos lankstumas;
- ✚ Galimybė gauti diferencijuotą informaciją;
- ✚ Pateikiamos informacijos išsamumas;
- ✚ Informacijos struktūra;
- ✚ Ryšio kokybė.

Nuo informacijos kokybės priklauso ir jos vertė, tačiau nereikia pamiršti, kad vertė susijusi ir su subjektyviais veiksniais.

1.2. DUOMENYS, INFORMACIJA, ŽINIOS

Informacinėje aplinkoje egzistuoja tokios sąvokos kaip *duomenys*, *informacija* ir *žinios*. Kad suvoktume šių sąvokų prasmės skirtumus, pateiksime jų interpretaciją remdamiesi *pranešimo* sąvoka [30].

Pranešimas yra grupė simbolių, kurie yra saugomi, apdorojami ir perduodami informacinėje organizacijos sistemoje. Kitais žodžiais tariant, organizacijos informacija yra saugoma, apdorojama ir perduodama kaip pranešimai. Pranešimo turinys, cirkuliuojantis informacinėje organizacijos sistemoje, turi skirtingus reikšmių lygius. Jie priklauso nuo to, ką pranešimas veikia – duomenis, informaciją ar žinias. Galima išskirti tris pranešimų reikšmių lygius, egzistuojančius informacinėje sistemoje (1.1. pav.).



1.1 pav. Pranešimų reikšmės lygiai

Duomenys – tai faktai apie stebimus tikrovės objektus, procesus, veiklą. Duomenys yra simbolių grupės, kurių pranešimo reikšmės lygmuo sąlygiškai žemiausias – tai pirminiai faktai ir pažiūros. Vykstant kai kuriems sprendimų priėmimo procesams duomenys naudojami tiesiogiai. Daugeliu atvejų duomenys yra grupuojami, įvairiomis formomis apibendrinami ir transformuojami, kad taptų *informacija*, naudojama priimant sprendimus, duomenis apibendrinančia pranešimo forma. Informacijos vertė sąlygiškai didesnė negu duomenų. Ji naudojama priimant sprendimus, vertinant realią situaciją. *Žinių* reikšmės lygmuo aukščiausias, kadangi jos vaizduoja žmogaus žinojimo veiksnius, samprotavimus, taisykles, kurie potencialiai galės būti panaudoti būsimose sprendimo situacijose.

Iš ateities informacinių sistemų tikimasi efektyvių priemonių, skirtų įgyti, palaikyti, išsaugoti, manipuluoti ir pateikti daug skirtingos informacijos. Naujos kartos sistemos reikalauja vartotojui palankios bendravimo aplinkos (sąsajos), samprotavimo galimybių ir diferencijuotos prieigos prie didelių informacinių bazių. Informacija šiuolaikinėje informacinių technologijų išsivystymo aplinkoje turi tenkinti bendrus reikalavimus:

- ✚ Informacija turi būti suprantama naudojant atitinkamą nuorodų struktūrą (kontekstą);
- ✚ Informacija turi būti tinkama tenkinti aktualius poreikius vykstant sprendimų priėmimo procesui;

- ✚ Informacija turi būti nauja ir veiksminga;
- ✚ Informacija turi skatinti (padėti) pasirinkti tinkamą sprendimą.

Veiksmai organizacijoje priklauso nuo atitinkamo organizacijos valdymo lygmens sprendimų priėmimo proceso rezultatų.

1.3. INFORMACIJOS VAIDMUO SKIRTINGUOSE VALDYMO LYGMENYSE

Informacines valdymo sistemas pagal atliekamas funkcijas galima suskirstyti į keletą architektūrinių lygmenų: organizacinės, valdymo, informacijos ir technologijų valdymo bei veiklos teisinio reglamentavimo struktūros [6]. Kiekvienas šių architektūrinių lygmenų turi tam tikro pobūdžio tarpusavyje susijusias funkcijas ir struktūrą.

Organizacinių informacinių valdymo sistemų lygmenį nusako:

- ✚ Organizacijos struktūra;
- ✚ Taisyklės, operatyvios duomenų kaupimo ir apdorojimo procedūros;
- ✚ Organizacijos funkcijos, pagrindinė veikla, gamyba, finansai, buhalterinė apskaita, išteklių paskirstymas;
- ✚ Išteklių planavimas ir valdymas, žmogiškieji ištekliai.

Kiekviena įstaiga, įmonė ar organizacija turi vidinę valdymo struktūrą, kurios kiekvienas struktūrinis elementas atlieka tam tikras funkcijas, yra pavaldus kitiems arba valdo žemesnio lygio struktūrinius komponentus. Tai organizacinės politikos dalis. Todėl informacinėse valdymo sistemose nustatant bendrą valdymo struktūrą, duomenų srautus, informacinius procesus į tai turi būti atsižvelgta.

Pagal organizacijos valdymo struktūrą bei hierarchiją galima išskirti kelis sistemos hierarchinius lygmenis, kuriuos dar praėjusio šimtmečio septintajame dešimtmetyje aprašė R. N. Anthony [2]. Kalbant apie sprendimų priėmimo sistemos ir hierarchinę organizacijos struktūrą reikėtų pažymėti skirtumus tarp kelių skirtingų sprendimų priėmimo funkcinių sričių: strateginio planavimo, vadovavimo valdymo, operacijų valdymo.

Strateginio planavimo ir valdymo srityje sprendimų priėmimas susijęs su tikslais ir jų pokyčiais, nustatant bendrą ir ilgalaikę institucijos veiklos politiką.

Vadovavimo valdymo srityje sprendžiami planavimo ir valdymo uždaviniai. Jie dažniausiai susiję su efektyvaus išteklių suradimo ir paskirstymo klausimais.

Operacijų valdymas turi garantuoti efektyvų operacijų atlikimą, remiantis specifinėmis taisyklėmis, pasirinktomis aukštesniuose lygmenyse.

Toks valdymo sričių suskirstymas buvo pavadintas Anthony taksonomija. Ji apibūdina skirtumus tarp struktūrinio ir nestrukūrinio problemų sprendimo.

Kiekvienas šių lygmenų skiriasi duomenų apdorojimo operatyvumu, kaupiamos informacijos apibendrinimo galimybėmis, veiklos ypatumais [20]. *Strateginės valdymo sistemos* yra ilgalaikės politikos, darbo organizavimo, planavimo, strategijų generavimo ir kitus strateginio valdymo uždavinius spręsti padedančios sistemos. Jos skirtos organizacijų vadovams sudaryti tiek trumpalaikes, tiek ilgalaikes prognozes, organizacijos veiklos, ir biudžeto paskirstymo planus bei numatyti ilgalaikius darbo jėgos pokyčius [13; 25; 20].

Aukščiausiojo lygmens vadovams pirmiausia rūpi strateginis planavimas ir kontrolė. Tad jiems reikalinga suderinta ir glausta informacija, kad būtų galima įvardyti, nustatyti arba parengti bendrąsias veiklos kryptis ir įvertinti bendrą veiklos lygį. Turimos informacijos nauda vertinama pagal jos kokybę, aktualumą, kiekybę ir tinkamumą valdyti. Nors šio lygmens strateginėms valdymo sistemoms išsamią informaciją teikia vadybos ir sprendimų palaikymo sistemos, planuojant strategiją svarbiausi yra išoriniai informacijos šaltiniai, teikiantys žinių apie ekonomines sąlygas, technologijos plėtotę, konkurentų veiksmus. Šią informaciją sunkiau surinkti ir kompiuterizuoti nei vidaus informaciją, nes papildomi duomenų šaltiniai organizacijai nepriklauso.

Taktinio valdymo lygmens informacinės sistemos skirtos vidutinio lygio vadovams, atsakingiems už strategijos, planų, tikslų, uždavinių vykdymą [13; 17; 20].

Taktiniam valdymui vykdyti kuriama keletas informacinių valdymo sistemų tipų. Galima būtų išskirti vadybos informacines sistemas, pritaikytas bendram organizacijos valdymui, darbo priemonių,

išlaidų kontrolei, personalo išdėstymo ir perskirstymo analizei atlikti. Sprendimų priėmimo ir paramos sistemos – tai patarinančios sistemos, kurios remdamosi informacijos analize palengvina sprendimų priėmimo procesus įvairiuose valdymo lygmenyse. Pavyzdžiui, tokios sistemos padeda vadovams priimti sprendimus dėl gamybos apimtys, realizacijos galimybių, atlikti pajamų ir išlaidų analizę.

Transakcijų lygmens informacinėmis valdymo sistemomis naudojamosi organizuojant ir kontroliuojant kasdienę veiklą, elementarias duomenų apdorojimo operacijas.

Transakcijų apdorojimo sistemos yra elementarių operacijų atlikimo ir jų panaudojimo sistemos (pvz., duomenų įrašymo, pakeitimo, ištrynimo, perdavimo ir panašių operacijų). Transakcijos atliekamos organizuojant ir kontroliuojant kasdienę duomenų apdorojimo veiklą.

Transakcijų valdymo sistemos užtikrina elementarių operacijų su duomenimis atlikimą (duomenų įvedimo, keitimo, panaikinimo operacijas). Šios sistemos skirtos rutininei veiklai, tokiai kaip pinigų pervedimas bei išmokos, kasdieninė medžiagų bei užsakymų kontrolė, darbuotojų darbo laiko apskaita, gamybiniai tvarkaraščiai ir kita. Atliekami informacinių sistemų uždaviniai yra konkrečiai apibrėžti ir sustruktūrinti. Atliktų transakcijų duomenys, tokių kaip medžiagų, finansų cirkuliacija, ataskaitos, pateikiami aukštesnio lygmens valdymo struktūroms.

Technologinę informacinių valdymo sistemų dalį sudaro:

- ✚ Įvairi elektroninė, kompiuterinė bei ryšio aparatūra (*Hardware*);
- ✚ Programinė įranga (*Software*);
- ✚ Informacijos saugojimo technologijos (*Storage*);
- ✚ Komunikacinės technologijos ir informaciniai tinklai (*Networks*).

Žinių lygmens valdymo sistemos parankios specialistams (inžinieriams, architektams ir kt.) bei biurų darbuotojams, kuriantiems naujas žinias ir dokumentus.

Žinių valdymo sistemų pagrindą sudaro žinių bazės ir įvairūs sprendimų priėmimo modeliai. Šios sistemos įgalina aprūpinti konkrečias darbo vietas. Pasitelkiant žinių sistemas kuriamos inžinerinės, administracinės, projektavimo, vadybos, analitinės, teisės patariamosios sistemos, kompiuterizuotos specialistų darbo vietos [10; 11; 18; 17; 29].

Šio lygmens informacinės valdymo sistemos skirtos kurti, organizuoti ir integruoti į organizacijos veiklą naujas žinias bei padėti organizacijai vykdyti šią veiklą intelektualią kompiuterinių sistemų priemonėmis. Šiuo metu tai greičiausiai besiplėtojančių ir populiarėjančių sistemų sritis.

Šios informacinės sistemos skirtos daugelio lygių organizacijų darbuotojams, žmonėms, sugebantiems kurti žinių potencialą (tai inžinieriai, teisininkai, gydytojai bei mokslininkai). Jų pagrindinė užduotis – kurti naują informaciją ir žinias. Taip pat žmonėms, naudojantiems sukurtąjį potencialą, kuriems svarbu įgyti daugiau žinių, apdoroti, bet ne kurti informaciją (pvz., sekretoriai, administratoriai, buhalteriai arba vadybininkai).

Darbuotojai, sugebantys sukurti žinių potencialą, dirba specializuotose žinių kūrimo sistemų darbo vietose, skirtose inžinieriams, architektams, vadybininkams ir pan.

Biurų automatizavimo sistemomis naudojamosi rengiant ir įforminant dokumentus, sudarant ir kontroliuojant darbo grafikus.

Informacinės valdymo sistemos, skirtos operatyvinei kontrolei atlikti, kasdien turi pateikti vadovui labai patikimą ir smulkią informaciją. Vadovas turi būti nuolat informuojamas, kokia organizacijos finansinė būklė, kiek ir kokių sunaudojama medžiagų ir t. t.

Biurų automatizavimo sistemose užtikrinama programinė aplinka, kai rengiami dokumentai, sudaromi ir tikrinami darbo grafikai, kalendoriniai planai, paskirstomos patalpos, organizuojami susirinkimai. Be paplitusios ir gerai žinomos tekstų apdorojimo programinės įrangos [27], naudojama ir sudėtinga dokumentų apdorojimo įranga, skanavimo ir archyvavimo bei darbo organizavimo sistemos. Dokumentai kompiuteriniais tinklais persiunčiami kitiems padaliniais, klientams, viršininkams bei archyvuojami kompiuterinėse laikmenose. Todėl kyla asmens duomenų apsaugos, perduodamų duomenų saugumo, dokumentą pasirašiusio asmens identifikavimo pagal skaitmeninį parašą problemų.

Vadybos informacinės sistemos sukurtos orientuojantis į vidinius organizacijos veiklos procesus [20]. Jos skirtos viduriniojo lygmens vadovams. Šio lygmens vadovams rūpi jų vadovaujamų skyrių veikla dabar ir ateityje, todėl jie turi žinoti svarbius įvykius, galinčius turėti įtakos jų vadovaujamiems skyriams. Vadybos informacinės sistemos parodo esamą bei buvusią organizacijos būklę ir yra skirtos

kontroliuoti darbo priemones bei išlaidas, taip pat analizuoti personalo paskirstymą ir persikirstymą. Jos paprastai fiksuoja ir apibendrina organizacijos savaitinius, mėnesinius ir metinius (o ne kiekvienos dienos) rezultatus bei įvykius. Rezultatai pateikiami įvairių ataskaitų pavidalu.

Sprendimų palaikymo sistemos skirtos įvairaus lygio vadovams, tačiau jose naudojama tiek vidinė organizacijos informacija, tiek duomenys iš kitų informacinių sistemų bei išorinių šaltinių [8; 20]. Tokios sistemos paprastai turi didžiules analitines ir modeliuojamąsias galimybes. Tai leidžia analizuoti duomenis ir informaciją, modeliuoti įvairius procesus ir priimti pusiau struktūrinius ir struktūrinius sprendimus. Šios sistemos paprastai dirba interaktyviu režimu: vadovas gali taikyti priartėjimo metodus, pateikti sistemai naujus klausimus, įvesti arba keisti duomenis ir žiūrėti, kaip valdymo sprendimas keičia galutinius organizacijos veiklos rezultatus.

Žinios, padedančios atlikti vykdomos veiklos įvertinimą ir kontrolę (pvz., diagnozuoti pažeidimus), apima daugelio skirtingų veiklos sričių duomenis, informaciją [8; 13]. Šiame procese svarbu nustatyti, kokie žmogaus žinojimo aspektai ir koku būdu įtraukiami į samprotavimo metodus ir problemos sprendimo strategijas [10; 11]. Skirtingų tipų dalykinės srities problemos paprastai reikalauja skirtingų strategijų ir samprotavimo metodų taikymo kompiuterinėje sistemoje.

1.4. DUOMENŲ STRUKTŪROS TAIKOMOSIOSE IR SISTEMINĖSE PROGRAMOSE

Daugelis realaus pasaulio komponentų kompiuterių sistemose turi būti išreiškiami per tam tikrus modelius, galinčius turėti matematinės išraiškos galimybes, grafines vaizdavimo notacijas, programinės įrangos organizavimo ir valdymo būdus. Programinės įrangos sistemose taikomos duomenų struktūros, leidžiančios išreikšti tokius kompleksinius elementus kaip aibė, eilė, sąrašas, sąrašas su rodyklės tipo kintamaisiais ir sąrašinėmis dinaminėmis duomenų struktūromis, dėklai (laikina informacijos saugykla) ir panašiai

[12]. Duomenų apdorojimo programų veiksmingumas ir aiškumas priklauso nuo duomenų saugojimo būdo. Uždavinį spręsti paprasčiau logiškai susietus duomenis saugant tam tikslui skirtose sudėtinėse duomenų struktūrose. Pagal savo savybes duomenys skirstomi į klases, vadinamas duomenų tipais: sveikieji skaičiai priklauso skaičių tipui, duomenys, turintys dvi logines reikšmes (*taip* ir *ne*, angl. *true* ir *false*), priklauso loginiam tipui, tekstiniai ar simboliniai duomenys priklauso simboliniam duomenų tipui (*Char* naudojamas žymėti tokio tipo duomenis Paskalio programavimo kalba ir *UML*, *Text* – duomenų bazių valdymo sistemoje *Access*).

Norint apibrėžti duomenų tipą patogiausia nagrinėti bendrąsias duomenų tipų savybes, būdingas daugeliui aukšto lygio programavimo kalbų, kurias suformulavo C. A. R. Horas (C. A. R. Hoare, 1975):

- ✚ Tipas apibrėžia klasę reikšmių, kurias gali įgyti kintamasis ar reiškinys.
- ✚ Kiekviena reikšmė priklauso vienam ir tik vienam tipui.
- ✚ Konstantos, kintamojo arba reiškinio tipą galima nustatyti iš konteksto arba iš paties operando pavidalo, neatsižvelgiant į reikšmes, gautas atliekant programą.
- ✚ Kiekvienos operacijos operandų ir rezultato tipai yra apibrėžti. Kai skirtingų tipų operacijos žymimos tais pačiais simboliiais (pvz., „+“ yra ir sveikųjų, ir realiųjų skaičių sudėtis), laikoma, kad toks simbolis yra daugiareikšmis ir žymi skirtingas operacijas. Kokią konkrečiai operaciją jis žymi, visada galima nustatyti transliuojant programą.
- ✚ Kiekvieno tipo reikšmių savybės ir su šiomis reikšmėmis vykdomų operacijų savybės apibrėžiamos aksiomomis.

Informacija apie tipus aukšto lygio kalbose leidžia išvengti beprasmių konstrukcijų programoje arba jas rasti ir nustatyti, kaip duomenys vaizduojami kompiuteryje ir kokie veiksmai su jais atliekami.

Matematikoje vartojami duomenų tipai: Dekarto sandaugos, žymėtosios sąjungos, aibės, funkcijos, sekos, rekursinės struktūros ir pan.



Duomenų tipai bei duomenų struktūrinimo būdai į programavimo praktiką atėjo kartu su aukšto lygio programavimo kalbomis.

Svarbu žinoti, iš kokių elementų sudaryti duomenys ir kaip jie išdėstyti atmintyje, t. y. kokia yra duomenų sandara, kokie veiksmai

atliekami su duomenimis. Svarbios yra leistinos manipuliavimo duomenų tipais ir jų apdorojimo funkcijos, todėl panagrinėsime pagrindinius ir svarbiausius struktūrinius duomenų tipus. Kad tinkamai pasirinktume duomenų struktūras ir priskirtume joms reikiamus duomenų tipus, svarbu išmanyti loginę ir fizinę duomenų organizaciją duomenų bazėse. Loginė duomenų organizacija – tai elementų išdėstymo tvarka, naudojama sudarant duomenų apdorojimo algoritmus, o fizinė duomenų organizacija – tai duomenų išdėstymo tvarka atminties įrenginiuose. Pagal duomenų aprašymo tikslumą priklausys ir jų apdorojimo algoritmų realizavimas programavimo kalbomis. Kad suprastume, kuo skiriasi duomenų modeliai (reliaciniai, tinkliniai), specialūs sistemos darbo su duomenimis programų paketai (duomenų bankai, duomenų bazių valdymo sistemos, žinių bazės), taip pat svarbu žinoti, kokiomis duomenų struktūromis jie gali manipuluoti.

1.4.1. Paprastieji, struktūriniai ir abstraktieji duomenų tipai

Pagal reikšmių struktūrinimo laipsnį duomenų tipai skirstomi į dvi klases:

-  Paprastuosius;
-  Struktūrinius.

Paprastųjų tipų reikšmės yra nedalomos, o struktūrinių tipų reikšmės yra sudarytos iš kitų – paprastųjų ar struktūrinių reikšmių.

Paprastieji duomenų tipai: simboliniai, sveikieji, realieji, loginiai, vardiniai, atkarpos, nuorodos.

Struktūriniai duomenų tipai: masyvai, įrašai, struktūros, Dekarto sandauga, alternatyva (junginys), aibė, bylos (komponentų sekos), rekursyvosios struktūros.

Abstraktieji duomenų tipai, kaip galima spręsti iš pačios sąvokos, yra tipai, kuriuose atsiribojama nuo dalies jų savybių, šiuo atveju – nuo jų reikšmių atvaizdų.

Patys paprasčiausi duomenų saugojimo ir apdorojimo būdai naudojami tada, kai duomenys surašomi į tiesines duomenų struktūras. Tiesinę struktūrą galima įsivaizduoti kaip aibę elementų, sujungtų į vieną eilę.

Aibė – tai to paties tipo logiškai susijusių objektų rinkinys.

Masyvas – tai vienodo tipo kintamųjų rinkinys, vadinamas vienu vardu. Kiekvienam rinkinio kintamajam kompiuterio atmintyje yra skiriama vieta. Kintamieji vadinami masyvo nariais arba elementais, kiekvienas jų turi savo numerį arba indeksą.

Tekstams vaizduoti ir apdoroti yra numatomi keli duomenų tipai. Vienas jų – simbolinis (*Char*), apibrėžiantis vieno simbolio duomenį. Tačiau parankiau dirbti ne su pavienėmis teksto raidėmis, o su jų junginiais: žodžiais, sakiniais ar tiesiog simbolių rinkiniais. Tam tikslui yra naudojamos eilutės.

Eilutė – tai struktūrinis duomenų tipas, aprašantis simbolių seką.

Programavimo kalbose veiksmų abstrahavimo priemonės yra funkcijos ir procedūros. Jų vartotojams svarbu tik tai, kokie veiksmai atliekami su duomenimis, o ne kaip šie veiksmai atliekami.

Duomenų tipas gali būti apibrėžiamas savo duomenų reikšmių aibe ir operacijomis, atliekamomis su šiomis reikšmėmis.

1.4.2. Dinaminės duomenų struktūros

Loginė duomenų organizacija – tai elementų išdėstymo tvarka, naudojama duomenų apdorojimo algoritmuose, o fizinė duomenų organizacija – tai duomenų išdėstymo tvarka atminties įrenginiuose.

Duomenų išdėstymo būdų yra labai daug, o veiksmai su duomenimis, duomenų apdorojimo funkcijos yra kelios: paieška, įrašymas, šalinimas, pertvarkymas (rūšiavimas), suspaudimas ir pan.

Nuorodos tipas leidžia iš paprastųjų ir struktūrinių tipų konstruoti sudėtingas dinamines duomenų struktūras [12].

Rodyklė – tai statinis kintamasis, saugantis kokio nors atmintinės baito adresą.

Sąrašas – tai struktūra, kurioje duomenys surašyti tam tikra tvarka. Sąrašų pavyzdžiai: darbuotojų sąrašai, kuriuose nurodyti vardai, pavardės, pareigos, atlyginimas. Daugeliu atvejų viena sąrašo eilutė charakterizuoja objektą, kuriam aprašyti reikalingi keli duomenų tipai. Pavyzdžiui, darbuotojų sąrašo vardai ir pavardės priskiriami eilutės tipo kintamiesiems, o parduodamų prekių kiekiai – realiojo tipo kintamiesiems.

Kiekviena sąrašo eilutė paprastai pateikiama įrašų.

Sąrašas nėra vien tik paprastas jį sudarančių įrašų rinkinys. Įrašai privalo būti tarpusavyje susiję, kad būtų galima nustatyti jų eilės tvarką sąrašė. Todėl kiekviename sąrašo įrašė, vadinamame grandimi, turi būti nuoroda į tolesnę grandį.

Sąrašus galima sudaryti vartojant įrašų masyvus.

Dinaminis sąrašas sudaromas iš įrašų, kurių kiekvienas turi rodyklę į tolesnį sąrašo narį.

Dirbant su dinaminiais sąrašais reikia mokėti:

- ✚ Suformuoti sąrašą;
- ✚ Peržiūrėti (išspausdinti) sąrašą;
- ✚ Atlikti paiešką sąrašė (rasti reikiamą grandį);
- ✚ Įterpti naują grandį;
- ✚ Pašalinti grandį.

Tokios struktūros labai plačiai naudojamos ne tik taikomoiose, bet ir sisteminėse programose. Pavyzdžiui, dėkluose yra saugomi pertraukiamų procesų parametrai, organizuojami lokalūs duomenys. Eilės yra populiarūs pagalbinių (buferinių) atminčių, kuriose kaupiami iš lėtų įrenginių gaunami arba į juos siunčiami duomenys, organizavimo priemonė. Eilių sudarymas ir valdymas būdingas masinio aptarnavimo sistemose. Su sąrašo tipo struktūromis dirbama duomenų bazėse, skaičiuoklėse ir kitose programinėse sistemose.

Sąrašas gali turėti skirtingas organizavimo formas. Jas galima suskirstyti į tokias rūšis:

- ✚ Tiesinis dinaminis sąrašas;
- ✚ Dėklas;
- ✚ Eilė;
- ✚ Tiesinis dvikryptis sąrašas;
- ✚ Žiedinis sąrašas;
- ✚ Lizdinis sąrašas;
- ✚ Medžio formos sąrašas.

Su šiomis duomenų struktūromis operuoja daugelis taikomųjų programų, tokių kaip *Excel*, *Access* ir kt. Vykdam programąs nagrinėjami įvairūs duomenų struktūrų veiksmā, tokie kaip sąrašo formavimas, spausdinimas, elementų paieška, įterpimas, šalinimas.

1.4.3. Rodyklės

Kiekvienam programoje aprašytam kintamajam kompiliatorius skiria atminties lauką, kurį apibūdina tokie parametrai: *adresas*, *lauko dydis*, *saugomų duomenų tipas ir reikšmė* [12]. Kintamojo pavadinimas programos tekste nurodo, kad turi būti manipuluojama jo reikšmė. Apsiriboti manipuliavimu vien tik kintamųjų reikšmėmis galima tik paprastus skaičiavimo algoritmus aprašančiose programose.

Sudėtingesnėse taikomosiose ir sisteminėse programose, kuriose yra aktualūs racionalaus dinaminio atminties paskirstymo, dinaminį struktūrų sudarymo ir kompiuterio įrangos valdymo fiziniu lygmenimi klausimai, tenka manipuluoti ne tik atmintyje saugomomis reikšmėmis, bet ir adresais. Adresų reikšmėms saugoti skirtos rodyklės, kurių aprašų sintaksė yra

`<duomenų_tipas>*<kintamojo_vardas>`.

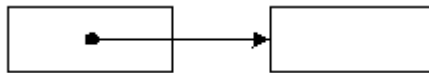
Programos tekste struktūra `*<vardas>` reiškia, kad turi būti vartojama rodyklės nurodomo lauko reikšmė. Rodyklėms gali būti suteikiamos tik to tipo objektų, kuriems apibrėžti jos buvo skirtos, reikšmės. Pavyzdys:

`int x, *px;`

`px = &x.`

Priskyrimo operatorius `px = &x` rodyklei `px` suteikia kintamojo `x` adreso reikšmę, todėl į `x` reikšmę galima kreiptis per vardą `x` ir struktūrą `*px`.

Rodyklė – tai kintamasis, kurio reikšmė yra elemento (kintamojo, masyvo, struktūros ar kt.) adresas.

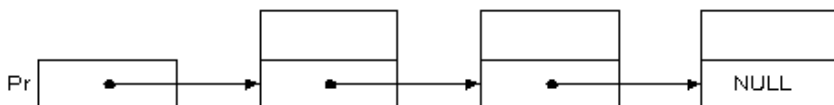


1.2 pav. Rodyklės žymėjimo pavyzdys

Programos darbo metu rodyklėmis susietus elementus galima šalinti, sukurti ir prijungti naujus. Tuomet užimamos atminties dydis padidėja arba sumažėja.

1.4.4. Dinaminis sąrašas

Dinaminis sąrašas – tai rodyklėmis susietų vienodo tipo elementų seka. Dinaminį sąrašą sudaro pradžios rodyklė ir vienodų dinaminio kintamųjų seka (1.3 pav.). Iš pradžios rodyklės paeiliui galima judėti per visus elementus nuo pirmo iki paskutinio, nes kintamieji yra susieti rodyklėmis. Paskutinio elemento ryšio lauke yra sąrašo pabaigos žyma – tuščia rodyklė (*NULL*).



1.3 pav. Dinaminio sąrašo žymėjimo pavyzdys

Čia elementas vaizduojamas kaip stačiakampis, padalintas į tiek dalių, kiek yra laukų elemento duomenų ir adreso struktūroje. Duomenų lauke saugomos reikšmės. Rodyklės tipo lauke saugomas adresas (nuoroda į kitą sąrašo elementą). Nuoroda pavaizduota linija su rodykle gale. Šios linijos pradžia yra nuorodos lauko viduje. Jeigu nuoroda neegzistuoja, tuomet linija nėra brėžiama ir laukas lieka tuščias. Tai reiškia, kad adreso reikšmė neapibrėžta. Realiose programose tokia situacija yra neleistina. Būtina užrašyti tuščio adreso reikšmę – konstantą *NULL*.

Tiesinio dinaminio sąrašo struktūra programoje gali būti aprašoma vienu arba dviem sakiniiais:

```
Struct List {
    int Sk;
    Struct List *next;
} *First
```

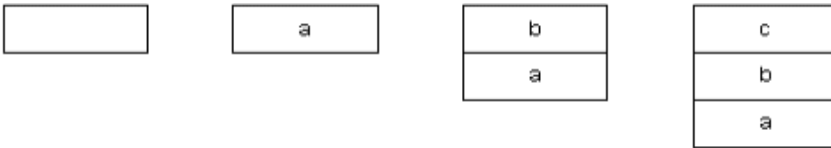
arba

```
Struct List {
    int Sk;
    Struct List *next;
};
List *First.
```


Su dinaminiais sąrašais gali būti atliekami tokie veiksmai kaip: sąrašo formavimas (keltis); sąrašo spausdinimas (išvestis); vieno ar kelių elementų paieška; naujų elementų įterpimas; vieno ar kelių elementų šalinimas; sąrašo tvarkymas ir kt.

1.4.5. Dėklas

Dėklas (*stack*) yra tokia duomenų struktūra, kai prieinamas tik vėliausiai įrašytas elementas. Programa gali perskaityti jo reikšmę ir pašalinti jį iš dėklo arba įrašyti prieš jį naują elementą. Dėklas yra paprasčiausiai ir lengviausiai tvarkomas tiesinis dinaminis sąrašas. Dėklo duomenų struktūrą apibūdina principas „paskutinis įeina, pirmas išeina“ (*angl. LIFO – Last In, First Out*). Grafiškai dėklas gali būti pavaizduotas vertikalia struktūra, kurios viršūnė yra piešinio viršuje.

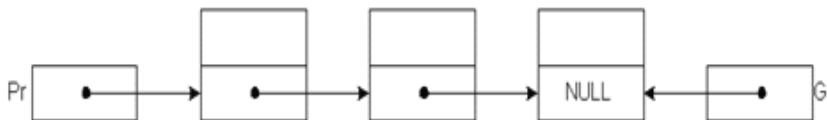


1.4 pav. Dėklo struktūra

Dėklas realizuojamas dinaminiais sąrašais arba masyvu rodykle nurodant paskutinį elementą (dėklo viršūnę). Norint dirbti su dėklu reikalingos dvi funkcijos: naujo elemento prijungimo prie sąrašo pradžios ir pirmojo elemento pašalinimo iš sąrašo (prieš tai šis elementas yra naudojamas atliekant veiksmus).

1.4.6. Eilė

Eilė – duomenų struktūra, kai nauji elementai įrašomi į galą, o esami elementai šalinami iš pradžios (1.5 pav.).



1.5 pav. Eilės žymėjimo pavyzdys

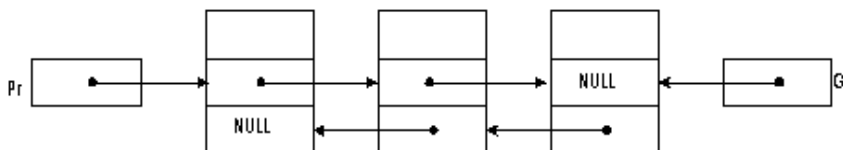
Eilės nuo dėklų skiriasi tik tuo, kad į jas duomenys yra siunčiami ir iš jų atrenkami priešinguose sąrašo galuose. Pertvarkant dėklą į eilę galima vieną jo tvarkymo procedūrą palikti, o antrąją reikia perrašyti atliekant rekursyvią sąrašo peržiūrą, t. y. ieškant eilės pabaigos. Suradus pabaigą, eilė papildoma tokiu pat būdu kaip ir dėkle.

Eilės pabaigos paieškai naudoti rekursyviojo arba ciklinio jos elementų perrinkimo būdą yra neracionalu. Toks paieškos būdas gali būti pateisinamas tik trumpose eilėse. Tvarkant ilgas eiles, jų elementų perrinkimo patartina vengti. Tai galima padaryti eilę papildant galinio elemento rodykle, kurios reikšmė būtų vartojama ir keičiama eilę papildant naujais elementais. Norint atlikti tokios eilės išorinį aprašymą rekomenduojama sudaryti dviejų rodyklių struktūrą ir ją tvarkyti modifikuotomis dėklo tvarkymo funkcijomis. Norint iš eilės padaryti dėklą, reikia sudaryti dar vieną funkciją, kuri leistų pašalinti galinius eilės elementus. Dėklas yra dviguba eilė, kurios elementai įrašomi arba šalinami ir viename, ir kitame gale.

Pagrindinis tiesinių sąrašų trūkumas yra tas, kad juose galima atlikti tik nuoseklų elementų perrinkimą. Dėl to sulėtėja juose esančių duomenų paieška, tampa neefektyvios su paieška susijusios tvarkymo operacijos. Siekiant išvengti šių tiesinių sąrašų trūkumų, yra sukurtos įvairios tiesinių sąrašų modifikacijos. Pavyzdžiui, tiesiniame sąrašo sudarant eilę, jis yra papildomas išorine rodykle, leidžiančia greitai papildyti sąrašą nauju elementu, įrašant jį į galą. Tokio paties poveikio galima pasiekti žiedinėje tiesinio sąrašo modifikacijoje. Tačiau abi šios modifikacijos nepalengvina paieškos pagal požymį ir galinio elemento šalinimo operacijų.

1.4.7. Dvikryptis sąrašas

Paieškai ir elementams šalinti labiau tinka dvikryptė tiesinių sąrašų modifikacija [12], kurios elementuose yra dvi rodyklės, rodančios pirmesnę ir tolimesnę iš gretimų elementų, ir viena arba dvi išorinės rodyklės (1.6 pav.). Tokiuose sąrašuose galima nuo bet kurio jų elemento prieiti prie visų kitų elementų.



1.6 pav. Dvikryptis tiesinis sąrašas

Dvikryptis sąrašas gali būti laikomas dviejų vienkrypčių sąrašų kompozicija, todėl jo tvarkymo funkcijos yra panašios į šių sąrašų tvarkymo funkcijas. Pagrindinis skirtumas yra tas, kad atliekant tvarkymo operacijas reikia formuoti dvi rodykles. Be to, galima sudaryti universalias elementų įterpimo ir šalinimo operacijas, kurios tinka visiems, ne tikai galiniams sąrašo elementams. Dvikrypčio dinaminio sąrašo struktūra gali būti aprašoma taip:

```
struct sar{
    int sk;
    sar * de;
    sar * ka;
};
```

Pradžioje sąrašas būna tuščias. Todėl pirmojo elemento rodyklės *Pr* (pradžia) ir galinio elemento rodyklės *G* (galas) reikšmės turi būti nulinės – *NULL*. Tuščias sąrašas nusakomas deklaruojant nulines pirmojo ir galinio elemento rodykles:

```
sar *P = NULL, *G = NULL.
```

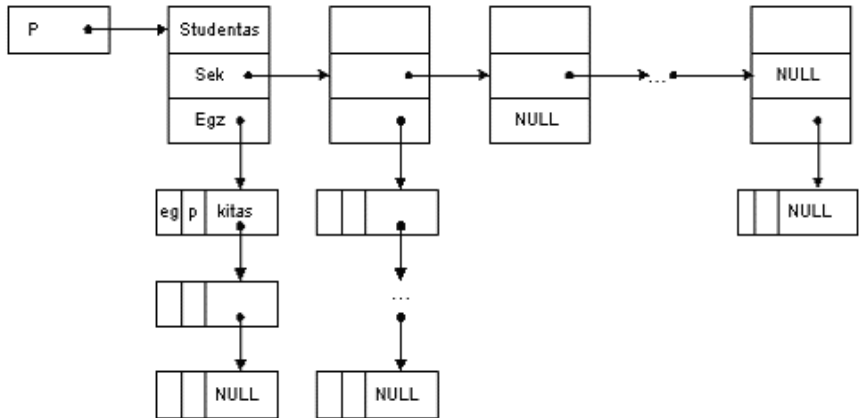
Dinaminio sąrašo elementas dažniausiai saugo tam tikro dydžio pagal kontekstą susietą informaciją apie vieną objektą. Sąrašo ilgis priklauso nuo objektų skaičiaus. Tačiau viename elemente galima saugoti informaciją apie kelis objektus, pavyzdžiui, studento sesijos pažymius. Toks dinaminis sąrašas vadinamas lizdiniu.

Lizdinis yra vadinamas sąrašas tokių elementų, kurių kiekviename telpa kelių objektų masyvas. Elementai vadinami lizdais. Lizdo pavyzdys pateikimas 1.7 pav.



1.7 pav. Lizdinio sąrašo pavyzdys

Kartais lizdą patogiau vaizduoti pasitelkiant sudėtingesnę dinaminį sąrašą. Tuomet lizdo masyvas keičiamas dinaminio sąrašo rodykle. Toks sąrašo pavyzdys pateikiamas 1.8 pav.



1.8 pav. Lizdinio sąrašo su sudėtingo dinaminio sąrašo elementais pavyzdys

Sąrašo srityje *Studentas* saugoma:

- Studento pavardė, vardas ir studijų knygelės numeris;
- Studento mokymosi vidurkis;

✚ Lauko *Sek* rodyklė nurodo kitą studentą;

✚ Lauko *Egz* rodyklė nurodo studento egzaminų sąrašą.

Egzaminų sąrašo elemento lauke *eg* saugomas egzamino pavadinimas, lauke *p* – įvertinimas, o lauke *kitas* – rodyklė į kito egzamino duomenis. Jeigu studentas dar nelaikė egzaminų, jų sąrašas bus tuščias ir rodyklės reikšmė bus *NULL*.

Tokia struktūra naudojama, kai nežinoma, kiek vietos reikės skirti tam tikriems duomenims. Darbas su tokiu sąrašu yra labai panašus į darbą su dviem tiesiniais sąrašais.

1.4.8. Medžio tipo struktūros

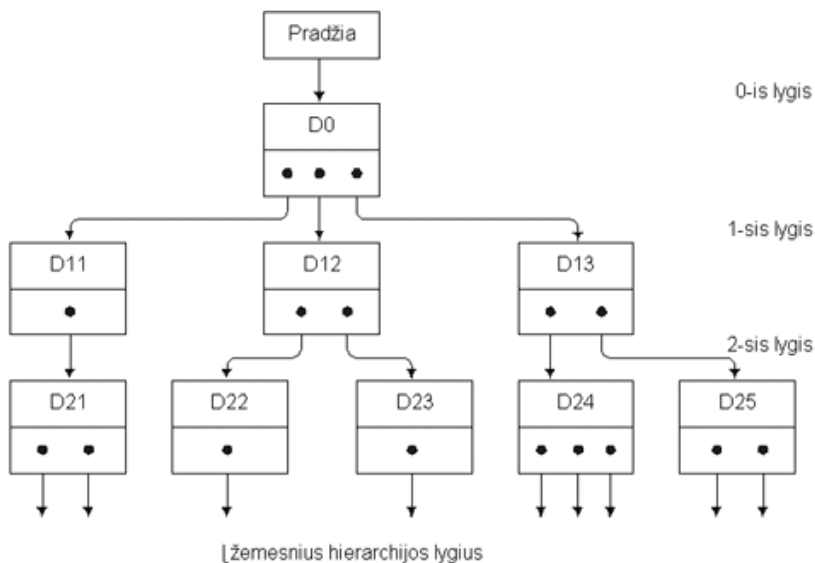
Sąrašas, kurio elementų ryšio tvarkos negalima išdėstyti vienoje grandinėje, vadinamas netiesiniu. Tai grafus vaizduojantys sąrašai. Čia viršūnės atitinka sąrašo elementų reikšmes, o viršūnes jungiantys lankai atitinka rodyklėmis aprašomus jų ryšius. Nuo grafo viršūnių sujungimo būdų priklauso jo savybės, grafo viršūnių perrinkimo būdai. Veiksmų su sąrašais sudėtingumas priklauso nuo grafų struktūros ir duomenų apdorojimo algoritmų.

Patys paprasčiausi ir dažniausiai naudojami yra medžio tipo netiesiniai sąrašai, kurių elementus galima išdėstyti į kelis hierarchijos (pavaldumo) lygius taip, kad pradiniam, nuliniame, lygyje būtų tik vienas elementas, o į kiekvieną hierarchijos lygio elementą vestų tik tai viena nuoroda ir tik iš aukštesnio hierarchijos lygio elemento (1.9 pav.).

Medžio tipo struktūras aprašant programomis, paprastai ribojamas nuorodų į žemesnius hierarchijos lygius (rodyklių) skaičius [12]. Jeigu medis apribojamas dviem nuorodomis, jis vadinamas binariniu.

Binarinių medžių elementai aprašomi taip pat kaip ir dvikrypių sąrašų elementai – ryšio dalyje formuojant dvi nuorodas. Tiktai priimta šias nuorodas žymėti kitokiais vardais, pabrėžiant, kad medžio šakos nukreiptos į dešinę arba kairę pusę (*de*, *ka*):

```
Struct medis {  
    void *data;  
    medis *ka;  
    medis *de;  
};
```



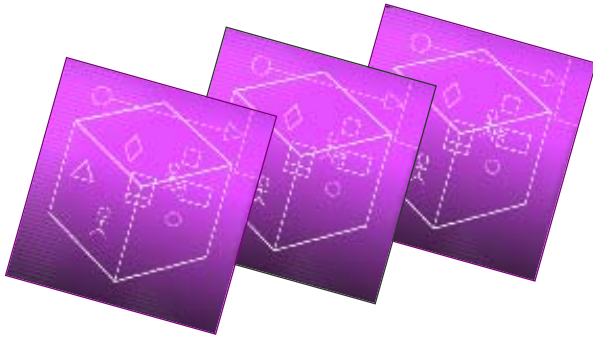
1.9 pav. Medžio tipo struktūros pavyzdys

Binariniai medžiai labai gerai tinka organizuoti tokioms duomenų saugojimo struktūroms, iš kurių dažnai tenka pagal požymius atrinkti pavienius elementus arba nedideles jų grupes. Aprašant medžio tipo struktūras taip pat reikia apibrėžti jų formavimo, tvarkymo bei apdorojimo procedūras.

Dažnai programose pasitaiko tarpusavyje susijusių kintamųjų ir jų reikšmių. Pavyzdžiui, datą galima apibūdinti trimis reikšmėmis – metais, mėnesiu, diena. Visus tris datos komponentus programose pavadinant skirtingais vardais galima jais operuoti kaip atskirais komponentais. Kai operuojame viena data, nepatogumų nejaučiame. Jų kyla norint skaičiuoti laiko intervalų trukmes.

2 skyrius

INFORMACINĖS SISTEMOS



- 2.1. Informacinių sistemų samprata
- 2.2. Informacinės sistemos gyvavimo ciklas
- 2.3. Informacinės sistemos analizė
- 2.4. Informacinės sistemos koncepcijos aprašymas
- 2.5. Objektinė informacinių sistemų projektavimo metodika
- 2.6. Administravimo informacinės sistemos projektavimas
- 2.7. Dalykinės srities duomenų srautų modeliavimas
- 2.8. Išskirstytosios informacinės sistemos
- 2.9. Informacinių sistemų kontrolė ir apsauga

Šiame skyriuje nagrinėsime *informacines sistemas*, jų projektavimo ypatumus, taikymo galimybes bei kūrimo klausimus. Pateikti universalią informacinės sistemos sąvokos apibrėžtį bus sunku, nes informacinių sistemų turinys, valdymo aplinka ir netgi esmė nuolat keičiasi. Tai priklauso nuo epochos, technologijų išsivystymo, inžinerinių sprendimų, organizacijų valdymo konteksto ir kultūros.

2.1. INFORMACINIŲ SISTEMŲ SAMPRATA

Šiuolaikines informacines sistemas nagrinėsime vertindami jų taikymo efektyvumą organizacijų valdymo, administravimo, verslo procesuose ir veikloje [11; 13; 14; 21; 22]. Šiuolaikiniuose vadybos procesuose, be *informacinės sistemos* sąvokos, vartojama ir *informacijos sistemos* sąvoka. Jų prasmė šiek tiek skiriasi.

Informacinę sistemą galima apibrėžti kaip *kompleksą komponentų, skirtų įvairių rūšių duomenims ir informacijai rinkti, saugoti, apdoroti, laikyti bei skleisti, siekiant tam tikrų organizacijos tikslų ir taikant kompiuterines technologijas*.

Informacinės sistemos neatskiriamos nuo technologijų ir techninės įrangos (kompiuterių, telekomunikacijų priemonių ir pan.). Jų pobūdį ir galimybes dažniausiai nulemia naudojamos technologijos. Tačiau pati technika savaime dar nėra technologija. Dabartinės informacinės sistemos įvaizdį galima sieti su kompiuteriu. Kompiuteris, kol su juo nėra susiejamos idėjos, tikslai, sprendžiamos problemos, uždaviniai ir tam tikri sugebėjimai, tėra metalo ir plastiko konstrukcija. Taigi minėtąsias kompiuterines sistemas galima laikyti technologijomis, kai jos tiesiogiai susiejamos su informaciniais tikslais, problemomis, poreikiais ir žmonių veikla.

Informacijos sistemų kūrėjai privalo būti susipažinę su taikomąja sistema (verslo, teisės, administravimo, technologijų ar kt.), kuriai bus skiriama informacijos sistema, ir atsižvelgti į vartotojų reikalavimus.

Nagrinėjant informacinę sistemą, būtina paminėti aplinką, kurioje ji funkcionuos. Organizacijos informacinė sistema kartu yra ir žmonių darbo įgūdžių, informacijos, informacinių technologijų ir veiklos procesų derinys. Su ja dirba žmonės – operatoriai, vartotojai

ir pan. Įdiegta informacinė sistema veikia ir formuoja reikiamą informaciją, pranešimus apie rezultatus. Veikiant informacinei sistemai, ji yra pastoviai aptarnaujama, sekamas jos darbas, šalinami aptikti jos komponentų defektai bei gedimai. Be to, ji turėtų būti nuolat tobulinama. Sistemai veikiant rekomenduojama pildyti jos eksploatacijos žurnalą, fiksuoti jame visus sistemos darbo sutrikimus ir visas vartotojų pareikštas pastabas apie sistemos darbą.

2.2. INFORMACINĖS SISTEMOS GYVAVIMO CIKLAS

Informacinė sistema kuriama laikantis tam tikrų tokioms sistemoms taikomų kūrimo principų pagal pasirinktus kūrimo ir gyvavimo ciklo modelius, skaidomus į etapus. Informacinių sistemų kūrimo ciklo modelių siūloma gana daug, pavyzdžiui, tradicinis, evoliucinis, greito prototipo sukūrimo, kaskadinis ir kt. [6].

Informacinės sistemos gyvavimo ciklas apima laikotarpį nuo šios sistemos sumanymo (konceptijos) iki veikimo pabaigos. Informacinės sistemos kūrimas – tai procesas, apimantis organizacijos tikslų, vartotojų poreikių bei galimybių analizę, siūlymo rengimą (sprendimų rengimą, įvertinimą, parinkimą, dokumentavimą), projektavimą, diegimą, palaikymą, tobulinimą. Formuluojuant sistemos kūrimo tikslus, konkretizuojami šių siekių įgyvendinimo kriterijai ir apribojimai. Jie formuoja sistemos funkcinę dalį, kurią sudaro kompiuterizuojamos valdymo funkcijos ir taikomieji kompleksai.

Siūlome panagrinėti tradicinį informacinės sistemos kūrimo ciklą.

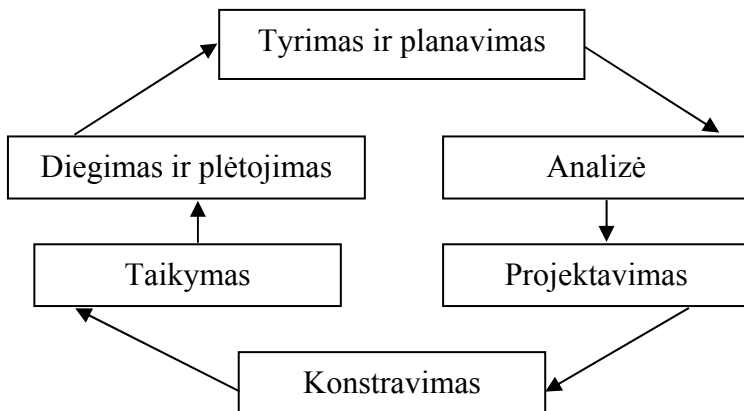
Tradicinis informacinės sistemos gyvavimo ciklas susideda iš šešių pagrindinių etapų. Jie gali būti skaidomi ir detalizuojami:

I. Tyrimas ir planavimas:

- 1.1. Sistemos nagrinėjimo poreikių nustatymas.
- 1.2. Pradinis tyrimas. Dalykinės srities ribų nustatymas.
Esamų sistemų tyrimas.
- 1.3. Informacinių ir komunikacinių technologijų pasirinkimas.

II. Analizė:

- 2.1. Problemos apibrėžimas.
- 2.2. Egzistuojančių sistemų ir technologijų įvertinimas.
- 2.3. Vartotojų reikalavimų ir naujos sistemos apribojimų nustatymas ir apibrėžimas.



2.1 pav. Tradicinis sistemos kūrimo ciklas

III. Projektavimas:

- 3.1. Konceptinis, loginis ir architektūrinis projektavimas.
Rekomenduojamo loginio modelio (konceptinio, loginio arba architektūrinio projekto) sudarymas.
- 3.2. Sistemos projektavimas (bendras projektavimas ir sistemos specifikacija).
- 3.3. Detalus (arba specifinis) projektavimas.
- 3.4. Fizinis projektavimas.

IV. Konstravimas:

- 4.1. Sistemos konstravimas.
- 4.2. Testavimas.
- 4.3. Instaliavimas, konvertavimas.
- 4.4. Derinimas.
- 4.5. Priežiūra įdiegus.

V. Taikymas:

- 5.1. Sistemos taikymas siekiant operatyviai įvesti konkrečios taikomosios srities duomenis.

- 5.2. Eksploatacija ir pokyčių įvertinimas.
- 5.3. Sistemos taikymas analogiško pobūdžio taikomosiose srityse.
- VI. Diegimas ir plėtojimas.
 - 6.1. Informacinės sistemos ir programų sistemos tiražavimas.
 - 6.2. Veikiančios sistemos naudojimas ir priežiūra.
 - 6.3. Sistemos kūrimas, plėtojimas ir tobulinimas.

2.3. INFORMACINĖS SISTEMOS ANALIZĖ

Informacinės sistemos analizės etapą galima skaidyti į smulkesnius etapus:

1. Struktūros analizės;
 2. Užduočių analizės;
 3. Sąveikos (bendravimo) analizės;
 4. Dokumentų analizės;
 5. Duomenų analizės;
 6. Procesų analizės.
- 1) Struktūros analizės metu nusakoma:
 - ✚ Organizacinė sistemos struktūra (kas veikia?);
 - ✚ Hierarchinė sistemos struktūra.
 - 2) Užduočių analizės metu nusakoma:
 - ✚ Kas yra atliekama (kokios operacijos yra atliekamos);
 - ✚ Kas atlieka šias operacijas;
 - ✚ Kada operacijos yra atliekamos;
 - ✚ Kodėl yra atliekamos operacijos.

Aprašant kiekvieną operaciją nustatoma:

 - ✚ Kokie duomenys ir informacija yra naudojami operacijai atlikti (pradiniai duomenys);
 - ✚ Kokie duomenys ir informacija gaunami atlikus operaciją (rezultatai);
 - 3) Sąveikos (bendravimo) analizės metu nustatoma:
 - ✚ Operacijų ryšiai;
 - ✚ Sistemos elementų sąveikos (bendravimo) ryšys.

4) Atliekant dokumentų analizę aprašomi dokumentai ir protokolai (ataskaitos), kurie bus pateikiami dalykinėje srityje.

5) Vykstant duomenų analizei aprašomos ir atskleidžiamos sudėtingų dalykinių sričių duomenų struktūros.

6) Atliekant procesų analizę pateikiama informacija apie tai, kokia tvarka bus atliekamos operacijos.

2.4. INFORMACINĖS SISTEMOS KONCEPCIJOS APRAŠYMAS

Informacinės sistemos projektuojamos laikantis tam tikros metodikos ir/arba metodologijos. Metodika nusako bendrus sistemos kūrimo principus, priemones, jų taikymo metodus. Per pastarąjį dešimtmetį viena iš dominuojančių sistemų kūrimo metodologinių krypčių yra vadinamoji struktūrinė sistemų analizė ir projektavimas. Struktūrinė sistemų analizė yra viena iš kertinių sąvokų, ji naudojama apibūdinant tam tikro pobūdžio sistemų kūrimo metodą. Nemažiau svarbi struktūrinės funkcinės analizės metodų kryptis.

Visi metodai, įgalinantys analizuoti, projektuoti, standartizuoti sistemos kūrimą, apibendrintai vadinami metodologija. Struktūrinės analizės metodas skirtas sistemos analizės ir projektavimo etapams. Struktūrinės sistemų analizės principai formaliai apibrėžiami kaip tam tikra metodologija [4; 30].

Kiekvienoje šių metodologijų turi būti keli pagrindiniai komponentai:

- ✚ Įrankiai, priemonės ir modeliai, skirti naujiems vartotojų reikalavimams aprašyti, analizuoti ir specifikuoti bei analizuoti ir įvertinti egzistuojančias sistemas.
- ✚ Sukurta tam tikro tipo bendros struktūros darbo aplinka, leidžianti nustatyti, kurie įrankiai, priemonės, modeliai ir formalizmai yra naudotini tam tikroje sistemos kūrimo proceso stadijoje ir kaip jie yra tarpusavyje jungiami.

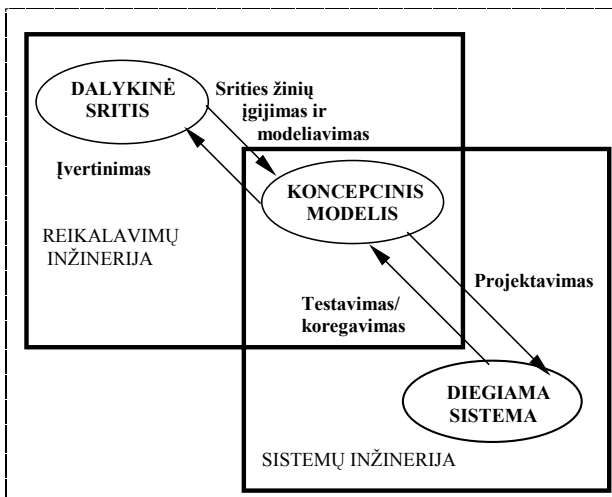
Informacinių sistemų projektavimo metodikai būdingi penki pagrindiniai aspektai:

- ✚ Priemonių ir technologijų aibės formavimas nustatant, kuri iš jų geriausia kiekvieno etapo situacijai valdyti.

- ✚ Darbo aplinkos kūrimo metodikos siūlymas laikantis standartų arba siekiant jų standartizuoti. Ši metodika nusako pagrindinius analizės ir projektavimo procesų etapus.
- ✚ Skirtumo tarp fizinio ir loginio sistemos požiūrių nustatymas ir jų naudojimas.
- ✚ Vartotojo vaidmens peržiūra sistemos kūrimo procese.
- ✚ Duomenų struktūrų, būtinų kaip propaguojamo projektavimo pagrindas, panaudojimas.

Sistemos projektavimo etape struktūrizuojama informacija, kuri bus svarbi organizacijos informaciniams objektams aprašyti ir vertinti. Pagrindinis projektavimo komponentas yra informacinės sistemos koncepcinis modelis, išreiškiantis sampratą, kaip sistema bus sudaryta ir kaip ji veiks (2.2 pav.). Čia nagrinėjami inžinerijos ir sistemų inžinerijos reikalavimų klausimai. Koncepciniam modeliui sudaryti gali būti taikomi įvairūs metodai, semantiniai modeliai, grafinių notacijų priemonės ir kt.

Semantiniai modeliai, susiję su abstrakcijos apibrėžimo mechanizmais – klasifikacija, agregacija, apibendrinimu ir asociacija, dalykinę sritį leidžia projektuoti ir nagrinėti kaip tikrovės modelį, tam tikrame abstrakcijos lygmenyje atsiribojant nuo konkrečių duomenų.



2.2 pav. Informacinės sistemos projektavimo etapai

Šiuo metu populiarūs objektinė projektavimo metodika [3]. Jos pagrindu sukurta unifikauta modeliavimo kalba *UML* tampa standartu aprašant sistemos struktūrą ir veikimo principus [6; 16]. Žinoma, egzistuoja ir kitos projektavimo metodikos, pavyzdžiui, *ARIS*, *ORACLE*, *KADS* ir pan. [4; 6].

Objektinės diagramų konstravimo priemonės (*technique*) skirtos objektų struktūroms ir elgsenai grafiškai vaizduoti ir jas specifiškai programavimo sistemose.

Sudarant koncepcinius modelius grafinės išraiškos priemonės vadinamos diagramomis. Projektuojant informacines sistemas naudojamos įvairios diagramų rūšys:

- ✚ Objektų (struktūrų) diagramos;
- ✚ Būsenų perėjimo diagramos;
- ✚ Sąveikos diagramos;
- ✚ Įvykių diagramos;
- ✚ Kontekstinės diagramos;
- ✚ KADS interpretacinės diagramos [4];
- ✚ Duomenų srautų diagramos.

2.5. OBJEKTINĖ INFORMACINIŲ SISTEMŲ PROJEKTAVIMO METODIKA

Sistema dažniausiai apibūdinama struktūros ir veiklos terminais. Veikla nurodo tai, kaip pasaulis keičiasi laike. Struktūra galima apibūdinti (specifikuoti) įvairias objektų konfigūracijas, pavyzdžiui, darbuotojai, dokumentai, projektai. Veiklos (elgsenos) specifikacija galima aprašyti tokius procesus kaip darbuotojo įdarbinimas, pranešimas apie darbuotojo pensijos amžių ir pan.

Objekto tipas yra idėja, samprata, kurią mes taikome įvairiems mūsų žinias sudarantiems dalykams arba objektams apibrėžti. Žmonės sąvokas taiko objektams apibūdinti. Terminas *objekto tipas* gali būti vartojamas ir vietoj sąvokos.

Mūsų tikrovė susideda iš realiai egzistuojančių objektų, kuriuos įvardijame sąvokomis, turinčiomis savo tipus. Sąvokų terminais mes kalbame apie konkrečius ir abstrakčius reiškinius, objektus, procesus, rezultatus ir pan. Kuriant kompiuterines sistemas svarbu,

kad mūsų kuriamas modelis kuo geriau atspindėtų tikrovę ir atitiktų objektus, kuriems mūsų sąvokos bus taikomos ir kuriems atspindėti skaitmeninėje erdvėje bus kuriami atitinkami objektai.

2.5.1. Objektų klasės

Objektą mes išskiriame pagal jo klasifikacinius požymius ir įvardijame tam tikra sąvoka (terminu) [3]. Ši sąvoka tarsi apibrėžia tam tikro objekto tipą, taikomą šios klasės objektams apibūdinti. Mes išskiriame stalų, asmenų, baldų, įmonių ir kitas objektų klases. Kiekviena ši klasė turi konkrečius objektus, t. y. klasės egzempliorius. Pavyzdžiui, apibūdinami *konkretų stalą*, kurio inventoriaus numeris 15, turėsime omenyje konkretų stalų klasės egzempliorių.

Klasė – aibė objektų, kuriuos sieja struktūros ir elgesio bendrumas. Bet kuris objektas yra klasės egzempliorius.

Tipas – tai tarsi reikalavimas, kurį turi tenkinti objektas, kad patektų į klasę. *Individualumas* – tai objekto savybės, išskiriančios jį iš visų kitų objektų. Klasė skirta atvaizduoti objektų rinkinį.

Apibrėždami sąvokas (konceptus, esybes) taikome:

- + *Klasifikaciją* – ja išskiriame objektų klases pagal jų skiriamuosius, prasminius bruožus;
- + *Apibendrinimą* – jo pagrindu apibendriname vieną objektų klasę kita, pavyzdžiui, *stalai* ir *kėdės* gali būti apibendrinami sąvoka *baldai*;
- + *Asociaciją* – ja nusakome ryšius ir apibūdiname, kaip objektai susiejami tarpusavyje;
- + *Agregavimą* – juo nusakome, iš kokių sudedamųjų dalių sudaryti objektai ir jų klasės.

2.5.2. Klasių dalijimas ir skirstymas

Sąvoka *dalijimas ir skirstymas* vartojama tam, kad galėtume aprašyti aibės padalijimą į nesutampančius poaibius. Padalytosios dalies viduje poaibiai negali sutapti. Tuo tarpu padalinių potipiai gali iš dalies sutapti.

Padalijimas gali būti traktuojamas kaip objekto tipo aibės suskirstymas/padalijimas į skirtingas nesutampančias grupes (potipius).

Pavyzdžiui, vienas būdas padalinti tipą *asmenys* yra suskirstyti jį į potipius *vyrai* ir *moterys*, kitas būdas – padalinti pagal tam tikrą amžių ir lytį.

Asmenų tipas dalijamas visiškai kitaip, kai atsižvelgiama į tai, ar asmuo tarnautojas ar ne.

Taigi kiekvienas objektų tipas gali būti padalintas/suskirstytas remiantis įvairiais aspektais.

Pavyzdžiui, asmuo gali būti vieno iš poaibių *vaikai*, *suaugusieji*, *pensininkai* egzempliorius. Tačiau tą pačią *asmenų* aibę dalijant pagal lytį, bus *moteriškosios* ir *vyrishkosios* giminės poaibiai. Rinkdami, kuriai grupei priskirti asmenį, vartojame *ARBA* tipo tvirtinimą. Galimybė sujungti keletą skirtingų idėjų, taikant *IR* tipo tvirtinimus gaunama, kai naudojamas daugybinis padalijimas. Kai tokios žinios yra automatizuojamos, semantikos aprašymo priemonės turi operuoti gerai apibrėžtu reikšmės tvirtinimu.

2.5.3. Objektų klasių ryšių tipai

Objektų tipų jungčių rinkinys formuoja tam tikros rūšies objektų tipą, vadinamą *ryšio tipu*. Ryšio (sąryšio) tipai ir žymėjimas yra priemonės, kuriomis aprašomos objektų jungtys (asociacijos).

Asociacijos ryšiai

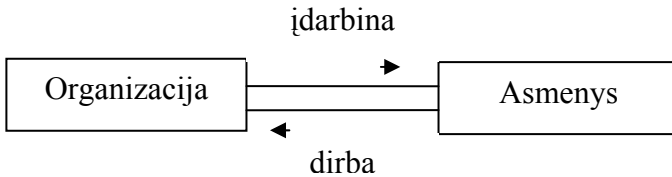
Asociacijos ryšių priemonės padeda suvokti, kaip prasmingai susieti įvairių tipų objektus.

Pavyzdžiui, nagrinėjame dvi objektų klases: *organizacija* ir *asmenys*, ir jų susiejimo ryšį, išreiškiantį asmenų įdarbinimą tam tikroje organizacijoje. 2.3 pav. iliustruojamas tokio tipo asociacijų ryšys ir jo grafinis vaizdavimas *UML* kalba [3].

Asociacija grafiškai žymima linija.

Stačiakampio formos viršūnės – gana tinkama vaizdavimo priemonė objektų klasei apibrėžti.

Objektų tipai apima objektų aibes. Asociacijos vaizduoja objektų aibių ryšius.



2.3 pav. Asociacijos ryšio tarp dviejų objektų klasių vaizdavimas

Pavyzdžiui, visi sąryšių egzemplioriai sudaro aibę ryšių, pažymėtų asmens įdarbinimo idėja, išreiškiamą asociacijos ryšiu.

Daugianaris (n narių) ryšys yra objektas, susiejantis kitus objektus, kai ryšio tipo egzemplioriai yra tam tikros n narystės. Daugianariškumo apribojimus išreiškia ryšio kardinalumo sąvoka.

Vaizdavimas pažymint (mapings) – tai sąryšiai, leidžiantys asociacijas, sąsajas traktuoti bendra prasme. Vaizdavimas nėra traktuojamas apskritai. Jis turi kryptį, kurią mes nusakome galvodami apie sąryšį. Organizacija įdarbina asmenis, o asmenys dirba organizacijoje.

Vaizdavimas dinaminiame pasaulyje tarsi nusako procesą. Praktiškai, kai nusakoma galimybė taikyti tokį vaizdavimą dinaminiame pasaulyje, jis reikalauja proceso.

Vaizdavimas, kurio metu visada kaip rezultatas pateikiamas vienas objektas, vadinamas vienareikšmiu.

Pavyzdžiui, kai norime, kad būtų pateiktas rezultatas apie asmens biologinę motiną, kiekvienas objektas iš objektų klasės *asmuo* vaizduoja vienintelį objektų klasės *moterys* objektą.

Kai vaizdavimas gali gražinti tam tikrą skaičių objektų, jis yra daugiareikšmis.

Pavyzdžiui, organizacijoje dirba daug asmenų, t. y. organizacijoje *Sauluva* dirba (yra įdarbinti) Martynas, Juozas, Dana ir kt.

Ryšio kardinalumo apribojimo samprata

Kardinalumo apribojimas nurodo minimalų ir maksimalų skaičių objektų, kurie gali būti gaunami kaip ryšio vaizdavimo rezultatas.

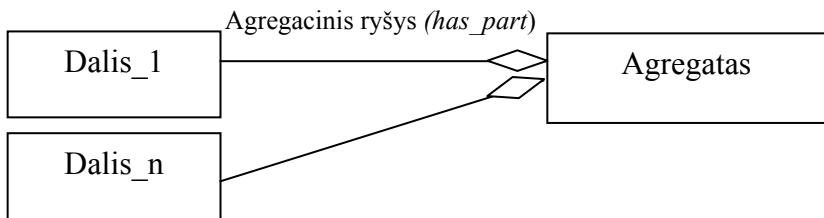
Kardinalumas nurodo skaičių, vaizduojantį objektų egzempliorių skaičių, gaunamą kaip ryšio rezultatą ieškant konkrečių klasių egzempliorių. Gali būti nusakomi tokie kardinalumo apribojimo tipai: 1:1 (vienas su vienu), 1:n (vienas su daugeliu), n:m (daugelis su daugeliu). Ryšių kardinalumo apribojimų žymėjimas taip pat gali įvairuoti.

Pavyzdžiui, asmuo gali dirbti keliose organizacijose (t. y. ryšio kardinalumo tipas bus 1:n)

Kompozicija (agregavimas)

Kompozicijos ryšiu nusakomas semantinis mechanizmas apibrėžia, kaip iš komponuojamų dalių sudaroma visuma. Kompozicija gali būti vadinama *agregavimu*. Agregavimo antonimas būtų dekompozicija.

Kompozicijos ryšiu galima apibrėžti objekto (daikto, reiškinių ar pan.) sudedamąsias dalis. Pavyzdžiui, burlaivis sudarytas iš atskirų dalių, tokių kaip valtis, burė, inkaras ir kt. Kompozicijos ryšys gali būti suprantamas daug konceptualiau ir subjektyviau, pavyzdžiui, kaip *santuoka* – vyras ir žmona, *komisija* – kompetentingų narių grupė.



2.4 pav. Agregavimo ryšio žymėjimas *UML* kalba

Objektų diagramos naudojamos parodyti objektų egzistavimą ir jų ryšius loginiame sistemos projekte.

Objektai sąveikauja per savo ryšius su kitais objektais, perduodamus pranešimus ir kitais bendravimo būdais.

Ryšys tarp dviejų objektų gali būti tada ir tik tada, jeigu yra asociacija tarp atitinkamų objektų klasių. Ši asociacija gali pasireikšti ir kaip kiti santykiai, pavyzdžiui, paveldėjimo ar turėjimo.

Asociacijos ryšio tarp dviejų klasių nustatymas reiškia kelią (ryšį) tarp šių klasių egzempliorių – šiuo keliu vienas objektas gali siųsti pranešimus kitam.

Apibendrinimas

Kad geriau suvoktume klasifikavimo ir apibendrinimo ryšius, panagrinėkime pavyzdį. Tarkime, mums reikia susisteminti ir surūšiuoti drabužių spintos turinį. Žiūrėdami į drabužių spintą mes matome, atpažįstame ir išskiriame (t. y. klasifikuojame) joje esančius objektus: paltus, lietpalčius, švarkus, marškinius ir t. t. Šią saugojimo vietą mes galime pavadinti paltų-lietpalčių-švarkų-marškinių... spinta. Kuo daugiau daiktų joje saugosime, tuo ilgesnis ir gremėzdiškesnis bus tokio tipo pavadinimas. Tačiau įvesdami į žodyną šiuos daiktus apibendrinančią sąvoką, galime supaprastinti ne tik pavadinimą, bet ir teisingiau išreikšti kai kuriuos semantinius mūsų aplinkos aspektus.

Apibendrinimas įgalina vartoti sąvokas, apibrėžiančias tai, kas bendra tokios klasės egzemplioriams. Taigi turėtume pagalvoti, kuri daug bendresnė sąvoka galėtų nusakyti tokias sąvokas kaip *paltas*, *lietpaltis*, *švarkas*, *marškiniai*? Žinoma, visas šias sąvokas galima būtų pavadinti *drabužiais*. Dar bendresnės sąvokos yra *prekės* arba *namų ūkio daiktai*, *materialūs daiktai* ir pan.

Apibendrinimas yra sąvokos išskyrimo veiksmas arba rezultatas (t. y. objekto tipas), kurio turinys visiškai apima vieną ar keletą kitų sąvokų.

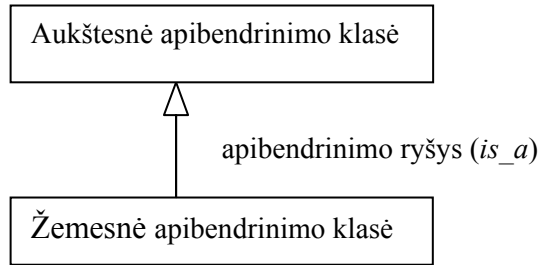
Apibendrinimas leidžia teigti, kad visi specifinio objekto tipo egzemplioriai yra taip pat daug bendresnio objekto tipo egzemplioriai, bet ne atvirkščiai.

Pavyzdžiui, *švarkas* ir *lietpaltis* yra *aprangos* objektai. Tačiau ne visi *aprangos* objektai yra *švarkai* ir *lietpalčiai*.

Apibendrinimu mes galime apibrėžti objektų tipų hierarchiją formuodami vis bendresnius objektų tipus.

Pavyzdžiui, *gyvybės forma* yra daug bendresnis objekto tipas (supertipas) negu *žmogus*, tuo tarpu *žmogus* yra bendresnė sąvoka negu *moteris*.

Apibendrinimo ryšiai diagramoje aprašomi kryptine linija su nenuspalvinta rodykle.



2.5 pav. Apibendrinimo ryšys

Objektai yra egzemplioriai, jie gali būti klasifikuojami. Klasifikacija ir apibendrinimas/specializacija yra skirtingi dalykai, turintys skirtingas hierarchijas.

Susipažinome su keliais skirtingais objektų hierarchijos aprašymo būdais:

- ✚ *Klasifikacija* – kai tam tikro tipo objektai yra jungiami į bendrą klasę (galima sudaryti jų poaibius, aibes, superaibes).
- ✚ *Apibendrinimu* – kai nustatomi tam tikri objektų tipų apibendrinimai (ši hierarchijos forma bus apibrėžiama *is_a* ryšiais tarp klasių).
- ✚ *Kompozicija (agregavimu)* – kai tam tikra visuma sudaroma iš tam tikrų sudedamųjų dalių (ši hierarchijos forma apibrėžiama *has_part* ryšiais tarp klasių).

2.6. ADMINISTRAVIMO INFORMACINĖS SISTEMOS PROJEKTAVIMAS

Šio skyriaus tikslas – susipažinti su informacinės sistemos projektavimo etapais, išsamiau panagrinėti informacinės sistemos, skirtos konferencijai administruoti, kūrimo etapus.

Nagrinėsime konferencijos organizavimo ir administravimo informacinės sistemos kūrimo pavyzdį. Siūlome administracinę sistemą projektuoti pagal objektinę projektavimo metodiką, taikant unifikuotą modeliavimo kalbą *UML* [3; 16].

Duomenų bazę siūlome kurti pasitelkiant reliacinio tipo duomenų bazių valdymo sistemas (DBVS), pavyzdžiui, *MS Access*.

2.6.1. Konferencijos organizavimo ir administravimo informacinė sistema

Apibrėžiant ir nusakant organizacijos informacinę sistemą turėtų būti nagrinėjami tokie klausimai:

- ✚ Konferencijos administravimo sistemos apibūdinimas;
- ✚ Tikslai;
- ✚ Informacinės sistemos kūrimo ciklas;
- ✚ Informacinės sistemos koncepcija (informacinių objektų aprašymo sistema);
- ✚ Sprendžiami uždaviniai;
- ✚ Duomenų srautų modeliai;
- ✚ Administracinės informacinės sistemos prototipo sukūrimas ir rekomendacijos bei kita.

Konferencijos administravimo sistema

Konferencijos dažniausiai organizuojamos pagal tam tikrą tematiką, atitinkančią kintamus besidominčių asmenų ir organizacijų probleminių klausimų ciklus. Konferencijos dažniausiai vyksta skirtingose geografinėse vietose, periodiškai renkantis šalį ir miestą, ku-

riame yra konferencijos tematiką reprezentuojančios ir dalyvaujančios šios srities darbe pagrindinės organizacijos ir mokslininkai.

Kuriant informacinę sistemą dažniausiai reikia taupyti jai skirtus ekonominius išteklius ir tenka kurti kompiuterinę sistemą, kuri būtų skirta ne vienai specifinei konferencijai administruoti, bet būtų pritaikoma daugeliui tokio pobūdžio uždavinių spręsti. Tad informacinės sistemos kūrėjų tikslas – kooperuotis su kitais šios srities darbuotojais ir kurti standartizuotą konferencijų organizavimo sistemą, kuri būtų visų naudojama tokio pobūdžio konferencijų darbui organizuoti. Matyt nesvarbu, ar tokią informacinę sistemą kursime tarptautinei, ar vietinei respublikinei konferencijai administruoti.

Kuriama kompiuterinė konferencijos administravimo sistema bus skiriama registruoti ir administruoti konferencijos dalyvius bei parengti konferencijos programą. Šioje dalyje nebus nagrinėjami tokie klausimai kaip konferencijos biudžeto tvarkymas ir apskaita, viešbučių kambarių rezervavimas, ir dalyvių laisvalaikio planavimas, konferencijos darbų spausdinimas.

2.6.2. Tikslų nustatymas

Vartotojai dažniausiai nori, kad sistema atliktų tam tikras funkcijas, todėl svarbu apibrėžti pagrindinius informacinės sistemos tikslus ir vartotojų reikalavimus. Pagal kontekstą nusakomi tokie reikalavimai kaip sistemos funkcionalumas, trumpai apibūdinama taikomoji sritis, technologijos, sąlygos, objektinė sistema ir darbų vykdymo eiga. Kiekvienas šis reikalavimas toliau bus detalizuojamas apibūdinant/aiškinant konkrečius sistemos kūrimo komponentus.

Pagrindiniai konferencijos administravimo informacinės sistemos reikalavimai pateikti 2.1 lentelėje.

2.1 lentelė. Konferencijos administravimo informacinės sistemos reikalavimai

Funkcionalumas	Sistema turės padėti administruoti tarptautinės konferencijos programą, dalyvių registraciją bei informacijos jiems suteikimą.
Vartotojai	Programinis komitetas, organizacinis komitetas, konferenciją administruosiantys vietinio personalo nariai, dalyviai.

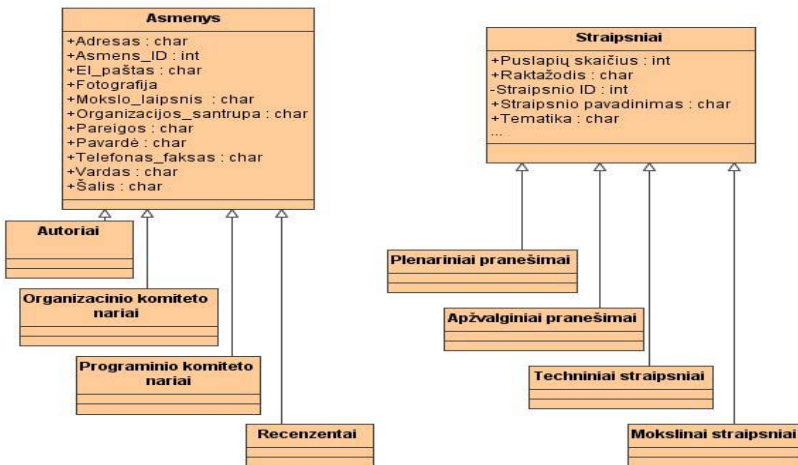
Technologijos	Vietinis kelių kompiuterių tinklas, serveris, internetas bei pastoviu režimu (<i>On-Line</i>) veikianči duomenų bazė.
Sąlygos	Užtikrinama tam tikra duomenų apsauga.
Objektinė sistema	Tarptautinės konferencijos rengimas registruojant dalyvius, straipsnius ir rengiant konferencijos programą.
Darbu eiga	Pirmiausiai sukuriamos administravimo priemonės, vėliau – asmenų, planuojančių ir rengiančių konferenciją, bendravimo aplinka, taip pat bendravimo su konferencijos dalyviais priemonės ir konferencijos programos vykdymo pagal numatytą darbų eigą planas.

Šie apibendrinti reikalavimai turėtų būti toliau detalizuojami kuriant informacinę sistemą.

2.6.3. Konceptinis projektavimas

Nagrinėjant konferencijos organizavimo informacinės sistemos projektavimo etapą aprašoma koncepcinė duomenų struktūros schema. Projektuojamosios objektų klasių struktūrų diagramos aprašomos renkant *UML* kalbos konstrukcijas.

Išskiriamos pagrindinės objektų klasės: *asmenys* ir *straipsniai*.



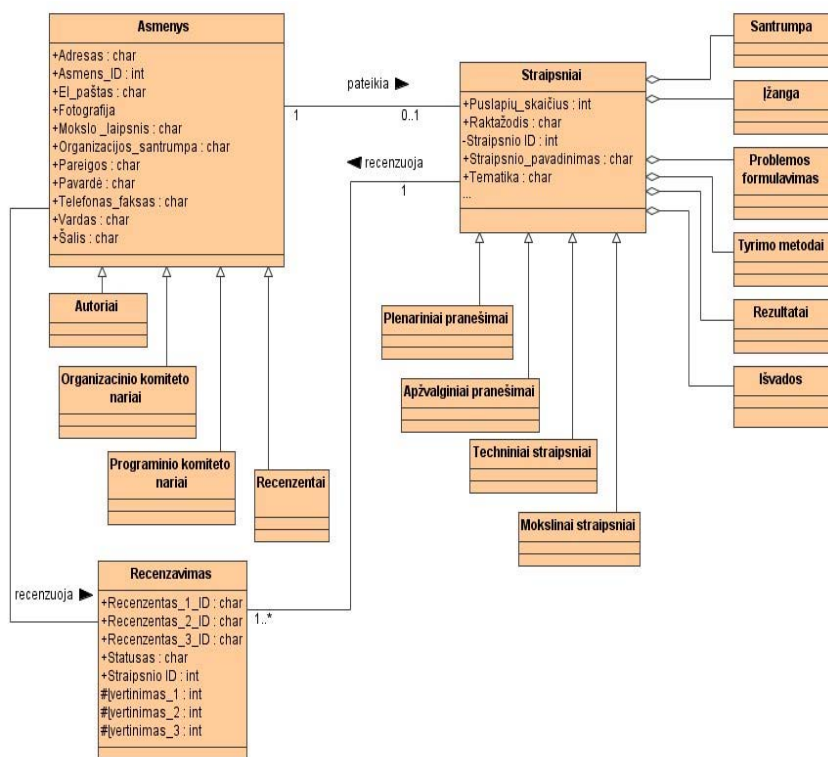
2.6 pav. Konferencijos administravimo informacinės sistemos pagrindinės duomenų klasės

Klases apibūdina jų parametrai, pavadinimai ir operacijos. *Asmenų* klasė apibrėžiama tokiais parametrais kaip *asmens_ID*; *pavardė*; *vardas*; *pareigos*; *organizacija*, *šalis* ir kt.

Išskiriamos apibendrinančios klasės, įvertinant konkrečius jų suskirstymo variantus ir diagramoje išreiškiant objektų klasių poabius, kurie bus svarbūs ir turės įtakos kuriant informacinės sistemos duomenų bazę.

Straipsniai šiame pavyzdyje apibendrina *plenarinių pranešimų*, *apžvalginių pranešimų*, *techninių pranešimų*, *mokslinių pranešimų* klases (2.6. pav.).

Schemoje nurodomi nagrinėjamoje dalykinėje srityje svarbūs agregavimo ryšiai.



2.7 pav. Konferencijos administravimo informacinės sistemos duomenų struktūros klasių diagrama

Pavyzdžiui, jei yra svarbu, iš kokių sudedamųjų dalių sudarytas straipsnis, agregavimo ryšiais diagramoje išreiškiame jo sudėtinę dalį: santrauką, įžangą, problemos formulavimą, rezultatus ir pan.

Taip pat nurodomi asociacijų ryšiai. Įvardytais asociacijų ryšiais diagramoje pažymime, kad *asmens* pateikia *straipsnius*, *straipsniai* yra recenzuojami ir pan. (2.7. pav.), apibrėžiame asociacijos ryšių kardinalumo apribojimus.

2.7. DALYKINĖS SRITIES DUOMENŲ SRAUTŲ MODELIAVIMAS





Dalykinę sritį (DS) galima nagrinėti daugeliu aspektų. Informaciniai DS modeliai vaizduoja informacinę DS struktūrą. DS duomenų srautų modeliai skirti modeliuoti DS vykstančią veiklą.

Dalykinėje srityje vykstančią veiklą galima skaidyti į duomenų, dokumentų, pinigų, gaminių ar kitokiais srautais siejamus procesus. Jeigu veikla yra reguliari, kiekvieną tokį procesą galima traktuoti kaip gaunamų srautų pertvarkymo (transformavimo) pagal tam tikras taisykles procedūrą. Procesų vykdymo taisyklės modeliuojamos pagal atitinkamus algoritmus.

Dalykinės srities srautams modeliuoti naudojama speciali koncepcinio modeliavimo kalba, vadinama duomenų srautų diagramų kalba [6; 30].

Duomenų srautų diagramomis galima aprašyti ir rankines, ir iš dalies kompiuterizuotas, ir visiškai kompiuterizuotas bei automatizuotas veiklas.

Duomenų srautų diagramų kalbos konstrukcijos yra šios:

-  Duomenų srautas;
-  Procesas;
-  Duomenų saugykla (duomenų bazė, informacinė sistema arba skaičiuoklėse saugoma duomenų struktūra);
-  Veiklos agentas.

Duomenų srautų diagramų kalboje *duomenų srautais* vadinami nustatytos struktūros duomenų perdavimo kanalai.

Duomenų srautų pavidalu vyksta duomenų mainai tarp procesų, duomenų bazių ir išorinių agentų. Jais gali būti perduodami ir

nevienalytės struktūros duomenų paketai. Komandas ir visus kitus valdančiuosius duomenis duomenų srautais perduoti draudžiama.

Duomenų srautai vaizduojami kryptį žyminčiomis rodyklėmis. Rodyklė rodo srauto kryptį. Ji žymima perduodamo srauto pavadinimu arba jo santrumpa.

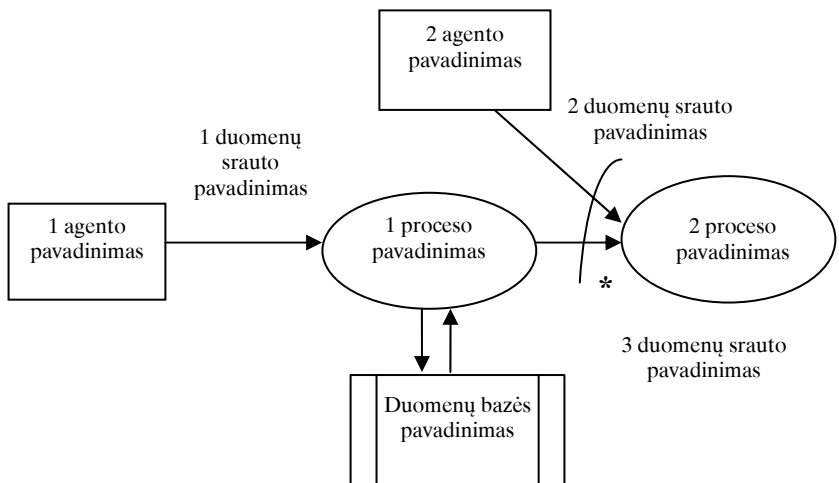
Duomenų srautų diagramų kalboje *procesais* vadinamos gautamų duomenų srautų transformacijos, atliekamos pagal iš anksto žinomas taisykles.

Procesas gali kurti naujus srautus, keisti gautų srautų struktūrą, kaupti tam skirtose saugyklose (duomenų bazėse) gautų srautų elementus arba koku nors būdu juos apdoroti.

Procesai vaizduojami paženklintais apskritimais, kuriuose įrašomas proceso pavadinimas arba jo santrumpa.

Duomenų srautų diagramų kalboje *veiklos agentais* vadinami modeliuojamos veiklos išorėje esantys reikalingų pradinį duomenų šaltiniai arba jos galutinių rezultatų vartotojai.

Agentai rodo, iš kur ateina pradiniai modeliuojamos veiklos srautai ir kur siunčiami šios veiklos rezultatai.



2.8 pav. Pagrindiniai duomenų srautų modelio komponentai

Duomenų srautų kalba turi ir tam tikrų procedūrinių elementų, t. y. srautų jungimo (konjunkcijos) ir išskyrimo (disjunkcijos) vaizdavimo galimybių.

Jeigu reikia pavaizduoti kelis vienam procesui perduodamus duomenų srautus, kurie turi ateiti arba būti išsiųsti tuo pačiu metu, juos reikia pažymėti konjunkcijos (*IR*) ženklu (susieję srautų linijas lanku jį pažymime*). Konjunkcijos ženklas jungia kelis duomenų srautus, perduodamus procesui arba išsiunčiamus iš jo vienu metu.

Skiriama griežtoji ir negriežtoji srautų disjunkcija.

Griežtoji srautų disjunkcija (*ARBA*) žymima pliuso (+) ženklu. Jeigu šis ženklas jungia kelis duomenų srautus, tai reiškia, kad vienu metu procesas gali priimti arba siųsti tik vieną iš srautų.

Negriežtoji srautų disjunkcija žymima # ženklu. Šis ženklas rodo, kad tuo pačiu metu procesas gali priimti arba perduoti bet koki srautų derinį.

2.7.1. Žodinis dalykinės srities aprašymas

Panagrinėsime, kaip sudaromas duomenų srautų modelis pareiškimų ir pranešimų apie nusikaltimus tyrimo dalykinėje srityje. Pirmiausia susipažinsime su veiklos etapais.

Policijos komisariato pareiškimų ir pranešimų apie nusikaltimus tyrimo procesai yra tokie:

✚ Pareiškėjai pateikia raštiškus pareiškimus/pranešimus apie nusikaltimus ir perduoda juos budėtojų skyriui (pareiškėjai gali būti fiziniai, juridiniai asmenys bei pareigūnai). Pareiškimai taip pat priimami paštu. Tokiu atveju jie atsiunčiami į komisariato sekretoriatą ir iš jo perduodami budėtojų skyriui.

✚ Visi gauti pareiškimai registruojami įvykių registracijos knygoje, t. y. registracijos žurnale, ir įrašomi į registracijos (apskaiotos) duomenų bazę.

✚ Pirminį įvykio tyrimą organizuoja budėtojas. Jis atlieka nukentėjusiojo pirminę apklausą, jeigu yra liudytojų, apklausia ir juos. Šio proceso metu surašomas paaiškinimų protokolai.

✚ Į įvykio vietą siunčiama operatyvinė tardymo grupė.

✚ Operatyvinė tardymo grupė ištiria įvykio vietą, surašo vietoje apžiūros protokolą: nubraižo įvykio schemą, fotografuoja, surašo patikslintos liudytojų apklausos duomenis. Apžiūros protokolo duomenys įrašomi į registracijos knygą.

✚ Policijos komisariato padalinio vadovas, susipažinęs su nusikaltimo bylos medžiaga, paveda įvykį ištirti (išnagrinėti) atitinkamam pareigūnui (tardytojui). Pavedimas vykdyti šį tyrimą ir pareigūno asmens duomenys įrašomi į registracijos knygą.

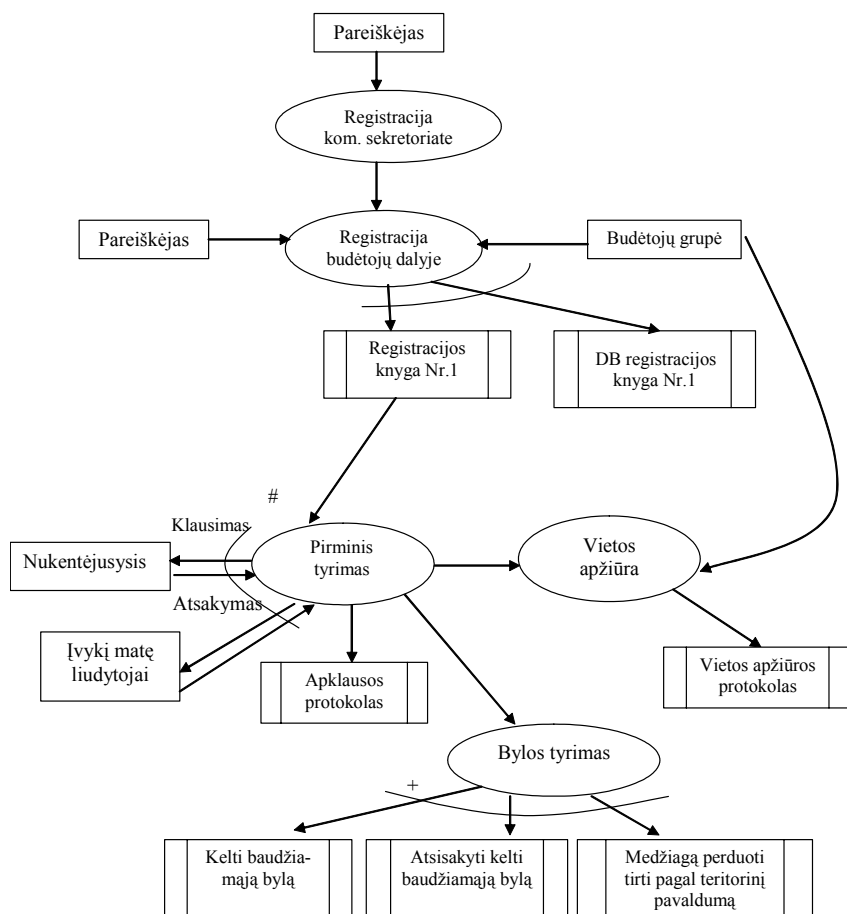
✚ Budėtojas informuoja pareigūną. Pareigūnas (tardytojas), gavęs pavedimą tirti pareiškimą, pasirašo registracijos knygoje, patvirtindamas, kad jį gavo. Per tris dienas arba vėliausiai per 10 parų tardytojas, analizuojantis bylos medžiagą, priima vieną iš šių sprendimų:

- Kelti baudžiamąją bylą;
- Atsisakyti kelti baudžiamąją bylą;
- Medžiagą perduoti tirti pagal teritorinį ar tarnybinį pavaldumą.

Priimtas sprendimas įrašomas į registracijos knygą.

2.7.2. Pranešimų apie nusikaltimus tyrimo duomenų srautų modelis

Pagal žodinį dalykinės srities aprašymą (2.7.1. skyrius) sudaromas duomenų srautų modelis (2.9 pav.).



2.9 pav. Pranešimų apie nusikaltimus tyrimo duomenų srautų modelis

2.8. IŠSKIRSTYTOSIOS INFORMACINĖS SISTEMOS

Didelės kompiuterinės sistemos projektuojant dažniausiai išskirstomos, t. y. informacijos apdorojimo procesai išdalijami (išskirstomi) į keletą kompiuterių [6].

Lokaliai veikiančios vadinamosios personalinės sistemos nėra išskirstomos ir yra suprojektuotos veikti viename personaliniame kompiuteryje ar darbo stotyje.

Įterptinės sistemos veikia viename procesoriuje arba integruotoje procesorių grupėje.

Išskirstytų sistemų programinė įranga veikia laisvai integruotoje grupėje kartu veikiančių procesorių, sujungtų į bendrą visumą tinklu.

Išskirstytosios sistemos pasižymi tokiomis savybėmis kaip: bendras išteklių naudojimas, jos yra gana atviros. Atsiranda galimybių daugelį informacijos apdorojimo procesų vykdyti lygiagrečiai. Šie procesai yra:

- ✚ Išteklių dalijimasis;
- ✚ Atvirumas;
- ✚ Lygiagretumas;
- ✚ Galimybė išplėsti sistemos architektūrą;
- ✚ Klaidų toleravimas.

Norint įdiegti tokias sistemas dažniausiai reikia daugiaprocesorinės, kliento-serverio, objektų skirstymo architektūros, objekto užklauso tarpininkų ir pagrindinių CORBA standartų. CORBA yra tarptautinis standartas – tarpinė programinė priemonė, skirta išskirstytų objektų bendravimui valdyti.

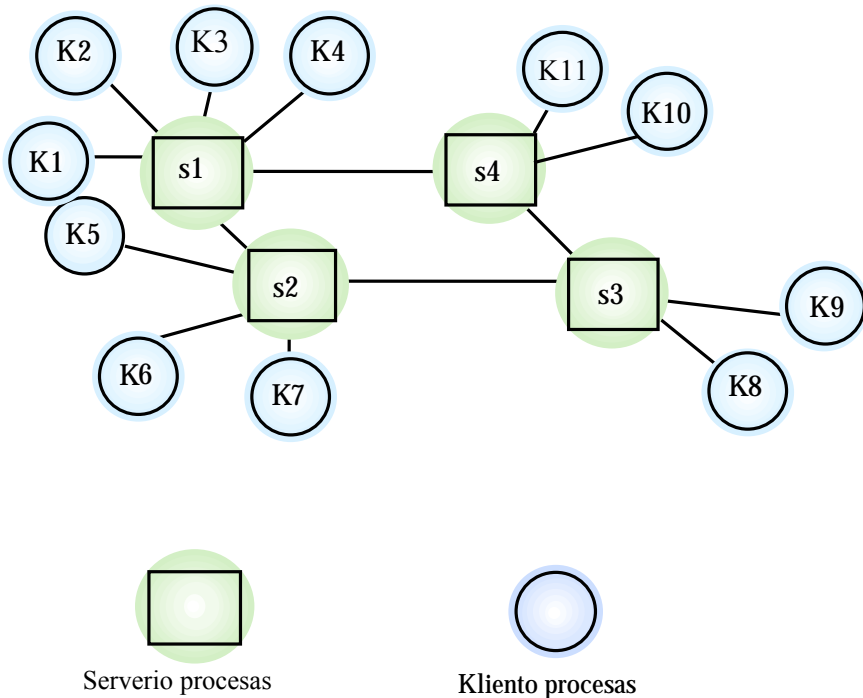
Išskirstytos sistemos architektūrinio projektavimo procesas apima sistemos struktūros kūrimo etapą, kai sistema suskaidoma į keletą pagrindinių posistemų ir yra nustatomos šių posistemų komunikavimo priemonės. Valdymo modeliavimas apima ryšių tarp skirtingų sistemos dalių valdymo modelius ir nustato modulinės sistemos dekompoziciją, kai nustatytos posistemės yra išskaidomos į skirtingose darbo stotyse veikiančius modelius.

Architektūrinis projektavimas apima ankstyvąją sistemos projektavimo proceso stadiją, palaiko ryšį tarp specifikacijos ir projektavimo proceso; jis dažnai vykdomas lygiagrečiai su kai kuriais spe-

cifikacijos veiksmais, kai nustatomos didesnės sistemos dalies komponentų identifikacijos ir jų bendravimo sąlygos.

Sistemos architektūra susijusi su programinės įrangos sprendimais, nustatančiais sistemos architektūrinį, valdymo ir posisteminių dekomponavimo modelius. Didelės sistemos retai kada paklūsta vienam architektūriniam modeliui. Sistemos dekompozicijos modeliai gali turėti saugyklų, kliento-serverių ir abstrakčius mašinų modelius. Valdymo modeliai susideda iš centralizuoto valdymo ir valdymo procesų modelių.

Srities architektūrinius modelius nusako modulinės dekompozicijos modeliai, susidedantys iš duomenų srautų ir objektų bendravimo modelių. Specialūs srities architektūriniai modeliai yra abstraktūs tam tikroje veiklos srityje veikiantys modeliai. Jie gali būti konstruojami abstrahuojant egzistuojančias sistemas arba idealizuojant nurodomus modelius.



2.10 pav. Išskirstytos kliento-serverio sistemos struktūros pavyzdys

Išskirstytųjų sistemų architektūros tipai gali būti grindžiami kliento-serverio arba išskirstytųjų objektų architektūra. Kliento-serverio architektūros sistemose paskirstytas paslaugas kviečia klientai. Paslaugas teikiantys serveriai traktuojami skirtingai nei klientai, kurie naudojami paslaugomis. Išskirstytųjų objektų architektūros sistemose nėra skirtumo tarp klientų ir serverių. Bet kuris objektas sistemoje gali tiekti ir naudotis kitų objektų paslaugomis.

Programinė įranga, valdanti ir užtikrinanti skirtingų išskirstytosios sistemos elementų darbą, yra sistemos viduje ir yra tarpinė sistemos darbo procesų dalis. Tarpinės programinės priemonės paprastai yra standartizuotos. Programinės įrangos pavyzdžiai gali būti transakcijas vykdydantys monitoriai, duomenų konverteriai, komunikavimo kontrolės priemonės.

Kliento-serverio architektūroje taikomosios programos yra sumodeliuotos kaip aibė serverių tiekiamų paslaugų ir kaip aibė klientų, naudojančių šias paslaugas. Klientai žino, kokie yra serveriai, bet serveriai nebūtinai žino, kas yra jų klientai. Klientai ir serveriai yra kaip loginiai procesai visoje sistemoje.

Taikomųjų programų architektūra turi tam tikrus lygmenis:

- ✚ Vaizdavimo lygmuo, skirtas sistemos vartotojų skaičiavimo rezultatams vaizduoti ir vartotojo duomenims įvesti;
- ✚ Taikomosios programos vykdymo lygmuo, skirtas programos specifinėms funkcijoms atlikti (pvz., tokios banko sistemos funkcijos kaip sąskaitos atidarymas, sąskaitos uždarymas ir t. t.).
- ✚ Duomenų valdymo lygmuo, skirtas sistemos duomenų bazėms valdyti.

Išskirstytųjų objektų architektūroje tarp klientų ir serverių nėra skirtumų, t. y. kiekviena skirstoma esybė yra objektas, teikiantis paslaugas kitiems objektams ir priimantis jas iš kitų objektų. Objektai bendrauja (komunikuoja) per tarpines programines priemones, vadinamas objekto užklauso tarpininkais (programinės įrangos magistralė). Išskirstytųjų objektų architektūrą projektuoti dažniausiai yra sudėtingiau nei klientų-serverių sistemas.

Išskirstytųjų objektų architektūros pranašumai pasireiškia tuo, kad sistemos projektuotojas gali atidėti sprendimą, kur ir kaip paslaugos turėtų būti teikiamos. Tai atvira sistemos architektūra, leidžianti pagal reikalavimus prijungti naujus išteklius, ji yra lanksti ir

išplečiama. Išskirstytų objektų architektūros sistemoje galima dinamiškai perskirstyti sistemos funkcijas ir objektus, migruojančius per tinklą pagal tam tikrus reikalavimus.

Išskirstytų objektų architektūra gali būti naudojama:

- ✚ Kaip loginis modelis, leidžiantis konstruoti ir organizuoti sistemą. Šiuo atveju yra galvojama, kaip pateikti programos funkcijas tiksliai išreiškiant paslaugomis ir paslaugų deriniais;
- ✚ Kaip lankstus metodas kliento-serverio sistemai įdiegti. Loginis sistemos modelis yra kliento-serverio modelis, bet ir klientai, ir serveriai yra realizuojami kaip išskirstyti objektai, susisiekiantys per programinės įrangos magistralę.

2.9. INFORMACINIŲ SISTEMŲ KONTROLĖ IR APSAUGA

Informacinių sistemų, kaip ir kitų kompiuterinių sistemų, atžvilgiu dažniausiai atliekama bendroji ir taikomoji kontrolė. Žinoma, slaptai informacijai saugoti reikia užtikrinti ir papildomas duomenų saugos priemones.

Bendroji kontrolė taikoma visoms kompiuterizuotoms sistemoms. Ji vykdoma jau sistemos projektavimo metu ir apima programinės bei techninės įrangos tikrinimą, garantuoja operacijų tikslumą, duomenų saugumą, administracinę tvarką. Jei bendroji kontrolė nėra taikoma, kompiuterinė sistema tampa nepatikima.

Įdiegimo kontrolė vykdoma įvairiuose taškuose, siekiant įsitikinti, kad sistemos kūrimo procesas yra tinkamai valdomas. Tikrinama programų kokybė, parengta dokumentacija. Sistemos galimybės įvertinamos pagal formalią kainos/naudos metodiką. Kontrolė suteikia pagrindą pritarti (arba ne) sistemos diegimui.

Sisteminių programų kontrolė apima operacinės sistemos dalis, kompiliatorius, failų, operacijų, įrašų tvarkymo ir kitų paslaugų programas. Minėtosios programos tiesiogiai tvarko duomenis ir jų failus.

Aparatūros kontrolė. Sistemos aparatūra turi būti fiziškai apsaugota ir prieinama tik asmenims, turintiems leidimus. Įėjimas į patalpas, kuriose yra kompiuterių, turi būti apribotas. Terminalai ir

mikrokompiuteriai, esantys kitose vietose, gali būti tik užrakinauose kambariuose. Kompiuterinė įranga turi būti apsaugota nuo gaisro, aukštos temperatūros, drėgmės. Jeigu kompiuterių darbas organizacijai ypač svarbus, numatomas jų avarinis dubliavimas, išigyjama atsarginių elektros energijos šaltinių.

Kompiuterio operacijų kontrolė padeda patikrinti, ar duomenų apdorojimo ir saugojimo operacijos vykdomos teisingai. Nenormalių operacijų baigmių atveju atliekamos atkūrimo procedūros, daromos atsarginės kopijos. Operacijų kontrolė apima ir rankines procedūras, skirtas surasti klaidas arba jų išvengti. Visi kompiuterių darbo sutrikimai turi būti fiksuojami ir analizuojami.

Duomenų apsauga užtikrina failų informacijos laikmenose saugumą. Leistinas tik sankcionuotas duomenų įvedimas, atliekamas dalinis įvedamų duomenų patikrinimas. Duomenų saugumas gali būti užtikrinamas keliais lygiais:

- ✚ Duomenų bazėmis gali naudotis tik asmenys, turintys leidimą;
- ✚ Tam tikromis programinėmis slaptumo priemonėmis yra apribojama prieiga prie kompiuterinių sistemų, saugančių svarbius duomenis (pvz., sisteminės programos tikrina slaptažodžius, identifikacinių kortelių duomenis ir pan.);
- ✚ Duomenų bazėms apsaugoti taikomiosiose programose naudojami papildomi slaptažodžiai.
- ✚ Siunčiami duomenys yra šifruojami, naudojami jų šifro raktai, kriptogramos.

Administracinė kontrolė – tai taisyklės, standartai ir procedūros, užtikrinančios tinkamą kontrolės sistemą. Svarbiausia – funkcijų atskyrimas, kontrolės procedūros, priežiūra.

Funkcijų atskyrimas yra fundamentalus bet kokios organizacijos vidaus kontrolės principas. Darbo funkcijos turi būti paskirstytos taip, kad kuo labiau sumažintų klaidų ir savanaudiškų manipuliacijų organizacijos turtu tikimybę. Asmenys, atsakingi už operacinę sistemą, negali inicijuoti operacijų, keičiančių duomenis. Atsakomybė už įėigą, apdorojimą, išėigą paskirstoma skirtingiems žmonėms. Informacinių sistemų skyrius yra atsakingas už duomenų ir programų failus, o vartotojai – už duomenų įvedimą ir klaidų taisymą.

Kontrolės procedūros apibūdina formalius informacinių sistemų operacijų tikrinimo standartus. Procedūros turi būti vadovų

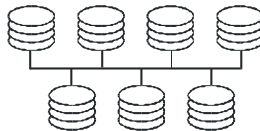
oficialiai patvirtintos. Jose turi būti aiškiai apibrėžiama atsakomybė ir atskaitomybė.

Personalo, dalyvaujančio atliekant kontrolės operacijas, priežiūra užtikrina numatytą darbų eigą. Priežiūra padeda pastebėti trūkumus, ištaisyti klaidas ir nukrypimus nuo standartinių procedūrų. Be tinkamos priežiūros net ir geriausia kontrolės sistema gali būti neveiksminga.

Taikomoji kontrolė kiekvienu atveju yra unikali. Kontrolė padeda užtikrinti konkrečios programos įvedamų duomenų tikslumą, išsamumą, apsaugą. Ją sudaro įvedimo, apdorojimo ir išvedimo operacijų kontrolė.

Kontrolės sistemos sudėtingumas, joje naudojamos procedūros turi atitikti duomenų svarbą. Pavyzdžiui, finansiniai duomenys apsaugos požiūriu yra svarbūs, o studentų mokymo duomenys – mažiau svarbūs. Etalonų duomenys yra svarbesni už konkrečių matmenų objektus. Nevienodos svarbos duomenims taikytinos skirtingos kontrolės procedūros.

Reta informacinė sistema naudoja visas minėtąsias kontrolės procedūras. Tai būtų brangu sukurti ir sudėtinga panaudoti. Sprendimas kurti kontrolės sistemą turi būti pagrįstas kainos/naudos analize, įvertinant klaidų pasirodymo tikimybę. Reikia parinkti tokias kontrolės procedūras, kurias pakankamai apsaugotą sistemą ir nesumažintų jos efektyvumo.



3 skyrius

DUOMENŲ BAZĖS IR JŲ VALDYMO SISTEMOS



- 3.1. Duomenų bazės samprata
- 3.2. Klasikiniai duomenų bazių modeliai
- 3.3. Reliacinių duomenų bazių projektavimas
- 3.4. Duomenų bazių valdymo sistemos
- 3.5. Pagrindiniai duomenų bazės kūrimo ir redagavimo etapai
- 3.6. Duomenų paieškos priemonės
- 3.7. Duomenų bazių praktiniai darbai ir metodiniai nurodymai
- 3.8. Informacijos pateikimo praktiniai darbai

3.1. DUOMENŲ BAZĖS SAMPRATA

Dažniausiai *duomenų bazė* suprantama kaip kartu saugomų ir susijusių duomenų, skirtų apdoroti kompiuteriu, visuma. Duomenys atitinka tam tikros realaus pasaulio dalies (taikymo ar dalykinės srities), skirtos kompiuterizuoti, modelį [1; 7; 21].

Kita duomenų bazės samprata yra platesnė, ji apima ne tik tvarkingai saugomus duomenis, bet ir programinę įrangą, skirtą duomenų bazėms kurti ir įvairiais būdais apdoroti.

Pagrindinis struktūrinis dalykinės srities vienetas yra esybė. Tai labai panašu į objektų klases objektinėje projektavimo metodikoje. Esybėmis vadinamos realaus pasaulio dalykų klasės, t. y. visa tai, ką galima apibūdinti bendriniais daiktavardžiais [6]. Projektuojant duomenų bazes aptinkamos fizinės ir koncepcinės esybės. Fizinės esybės – tai daiktai, reiškiniai, procesai, subjektai, agentai ir pan. Koncepcinės esybės – tai sąvokos, idėjos. Konkrečios esybės realizacijos vadinamos esybių egzemplioriais. Nagrinėjant pavyzdį apie asmenis, *asmuo* yra esybė, konkretus *asmuo Jonas Jonaitis* yra jos egzempliorius.

Kuriant informacines sistemas, dažniausiai domimasi ne visomis konkrečios srities esybėmis, o tik tomis, kurios yra svarbios nagrinėjimo tikslais. Dažniausiai analizuojamos pagrindinės esybių kategorijos:

- ✚ Asmenys (organizacijų darbuotojai, klientai, tiekėjai ir pan.);
- ✚ Buvimo vietos (organizacijos, paslaugų tiekimo, gamybos, saugojimo vietos);
- ✚ Kitos materialios esybės (gaminiai, nuosavybė, inventorius, finansinės sutartys, pajamos);
- ✚ Teisinės ir kitos nematerialios esybės (veiklą reglamentuojantys teisiniai aktai, intelektinė nuosavybė, organizacijos padalinių struktūra).




Duomenų bazių pavyzdžiai: konferencijos dalyvių ir pateiktų straipsnių duomenų bazė, bibliotekos kartoteka, telefonų abonentių knyga, pirkėjų užsakymų registracijos žurnalas, gyventojų registras ir kita.

Duomenys saugomi ir apdorojami ne bet kaip, o laikantis tam tikrų susitarimų, taisyklių. Kitaip tariant, duomenys duomenų bazėje yra sutvarkyti tam tikru būdu.

Duomenų bazė – tai struktūrizuota duomenų saugykla, kurioje saugomi duomenys ir informacija apie realius modeliuojamos dalykinės srities objektus, procesus, įvykius arba reiškinius. Duomenų bazė turi būti organizuojama taip, kad būtų patogų apdoroti ir valdyti tiek bendrą šios informacijos visumą, tiek atskiras jos dalis.

Reiktų skirti vadinamąjį fizinį ir loginį duomenų organizavimą. Pirmasis nurodo duomenų fizinio išdėstymo kompiuterio atmintyje būdus, o antrasis – duomenų struktūros vaizdavimą – modelį, reikalingą vartotojams, projektuotojams.

Kuriant duomenų bazes reikia laikytis tam tikrų reikalavimų:

-  Duomenys turi būti integruoti;
-  Duomenys turi būti organizuoti vengiant jų pertekliaus;
-  Turi būti laikomasi nepriklausomybės principo.

Duomenų integruotumas reiškia, kad visi duomenys kaupiami ir saugomi kartu, nustačius jų tarpusavio ryšius. Taip saugomus duomenis dažniausiai naudoja ne vienas, o keli vartotojai.

Nepertekliaus savybė nusako duomenų organizavimo principus, kai duomenys yra struktūrinami ir saugomi vengiant jų dubliavimo, kartojimosi. Šių tikslų pasiekama tinkamai projektuojant loginį duomenų bazės modelį, kai duomenys išskaidomi į atskiras lenteles, taikant duomenų normalizavimo principus. Šie principai nusako duomenų struktūrinimo būdus, leidžiančius sutvarkyti duomenų bazę iki reikiamos normalinės formos. Išskiriamos pirma, antra, trečia ir ketvirta (I, II, III ir IV) normalinės formos. Normalizuojant duomenis jie skaidomi į lenteles, lentelėms priskiriami duomenis siejantys papildomi laukai.





Turinti perteklių duomenų bazė yra neefektyvi, per didelė, operatyvioji paieška trunka daug ilgiau. Kai sukuriamos kelios duomenų kopijos, joms veltui eikvojama atmintis, o modifikuojant duomenis tenka kelis kartus naudoti tas pačias atnaujinimo operacijas. Be to, kai duomenų kopijos atitinka skirtingas atnaujinimo stadijas, tai gali lemti prieštaringos informacijos pateikimą.

Duomenų nepriklausomybė reiškia, kad modifikuojant duomenis jų apdorojimo taikomosios programos nesikeičia.

Koncepcinis modelis projektuojamas neatsižvelgiant į konkrečių duomenų ir programinės įrangos versijas.

3.2. KLASIKINIAI DUOMENŲ BAZIŲ MODELIAI

Pagal tai, koku būdu duomenys jungiami į visumą, išskiriami pagrindiniai, šiuo metu vadinami klasikiniai, duomenų organizavimo (tvarkymo) modeliai: reliacinis, hierarchinis, tinklinis. Populiarus tampa objektinis duomenų modeliavimas. Atsižvelgiant į tai, koks modelis naudojamas organizuojant duomenis, galima išskirti ir duomenų bazių rūšis:

-  Reliacinės duomenų bazės;
-  Tinklinės duomenų bazės;
-  Hierarchinės duomenų bazės;
-  Objektinės duomenų bazės.

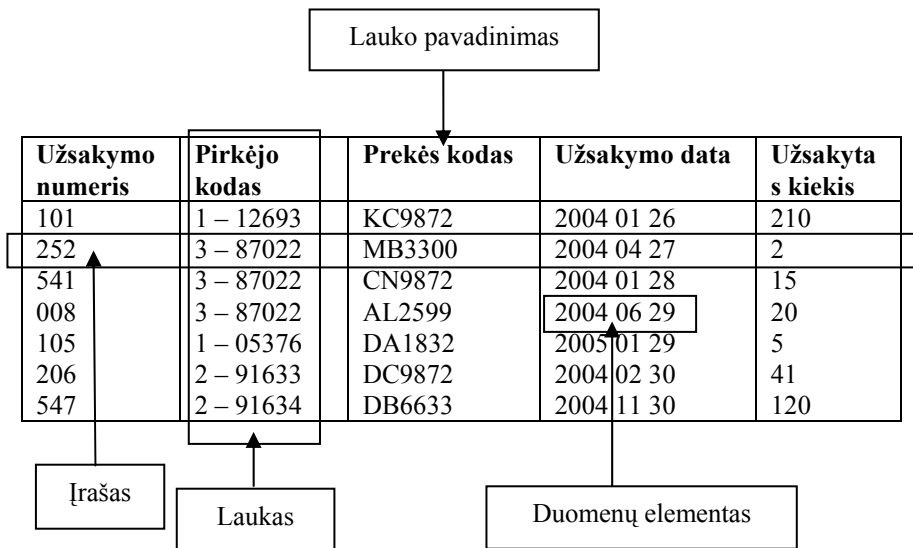
Pastaruoju metu bene populiariausios duomenų bazių valdymo sistemos, kuriose duomenys organizuojami pagal reliacinį modelį. Tokios bazės vadinamos reliacinėmis.

3.2.1. Reliacinės duomenų bazės

Reliacinės duomenų bazės – tai duomenų bazės, kuriose modeliuojamos dalykinės srities esybės išreiškiamos lentelėmis ir jų ryšiais.

Reliacinė duomenų bazė yra tokia duomenų visuma, kurioje informacija (duomenys) saugoma dvimatėse lentelėse [1]. Lentelę sudaro eilutės ir stulpeliai, taigi lentelės forma yra įprasta ir suprantama vartotojui. Lentelės eilutės vadinamos įrašais, jie atitinka konkrečias objektų klasės egzempliorių parametrų reikšmes. Stulpeliai vadinami laukais (kortedžais, domenais). Laukams suteikiami pavadinimai (pvz., *Užsakymo numeris*, *Pirkėjo kodas* ir pan.), atitinkantys objektų klasės parametro semantinę prasmę. Reliacinės duomenų bazės modelis turi tenkinti reliacinės algebros reikalavimus.

Bet kurios eilutės ir bet kurio stulpelio sankirtoje įrašomas konkretus duomuo, atitinkantis jo semantiką (t. y. prasmę). Stulpelio ir eilutės sankirtoje (ląstelėje) gali būti įrašyta tik viena duomenų elemento reikšmė, o ne reikšmių rinkinys.



3.1 pav. Užsakymų registracijos žurnalo duomenų bazės elementai

Reliacinėse duomenų bazėse kiekviena lentelė turėtų būti kuriama laikantis tam tikrų reikalavimų:

- ✚ Laukuose turi būti saugomi vienaarūšiai duomenys, t. y. kiekvieno lauko reikšmės turi būti vieno tipo. Tačiau skirtinguose laukuose gali būti įvairių tipų duomenys.
- ✚ Visi įrašai turėtų būti vienodai organizuoti.
- ✚ Lentelėje negali būti tuščių arba identiškų įrašų, t. y. įrašų, kurių visi duomenys kartojasi, nors atskiri duomenų elementai gali būti tušti arba kartotis.
- ✚ Įrašų ir laukų išdėstymo tvarka lentelėje nėra svarbi.

Atliekant duomenų apdorojimo operacijas lentelės eilutės ir stulpeliai gali būti peržiūrimi bei redaguojami bet kokia tvarka.

Lentelės pavadinamos prasminiais vardais (arba jų trumpiniais). Laukų, priklausančių pirminiam raktui (raktai yra laukai arba laukų grupės, kurių įgyjamos reikšmės nesikartoja), vardai pabraukiami. Toks užrašas vadinamas lentelės schema. Pavyzdžiui, *Užsakymo* registracijos lentelės schema gali būti aprašoma taip:

Užsakymai (užsakymo numeris, pirkejo kodas, prekės kodas, užsakymo data, užsakytas kiekis).

Reliacinėje duomenų bazėje lentelės gali būti siejamos tarpusavyje. Ryšį tarp atskirų lentelių nustato bendri, sutampantys tų lentelių laukai, dar vadinami siejančiais laukais. Taip susietų lentelių visuma ir apibrėžia reliacinį modelį. Kitaip negu rodykliniuose duomenų modeliuose, kuriuose ryšiams iliustruoti naudojamos rodyklės, reliaciniame modelyje nėra atskirų elementų ryšiams vaizduoti – jie išreiškiami lentelėse (siejančiais laukais ir jų ryšiais). Šie siejantys laukai tampa pertekliniais, tačiau jie yra reikalingi. Pertekliniai laukai dažniausiai įvedami dėl dviejų priežasčių:

- a) Daliai lentelių reikalingi laukai, skirti suformuoti pirminius raktus, t. y. vienareikšmius lentelių įrašų identifikatorius;
- b) Kai kurioms lentelėms reikalingi laukai, kurie nėra pirminiai raktai (ar jų dalis), bet naudojami nustatyti ryšiams.

Pagal ryšio kardinalumo apribojimų rūšis skiriami trijų rūšių ryšių apribojimai – vienas su vienu (*one-to-one*), vienas su daugeliu (*one-to-many*), daugelis su daugeliu (*many-to-many*).

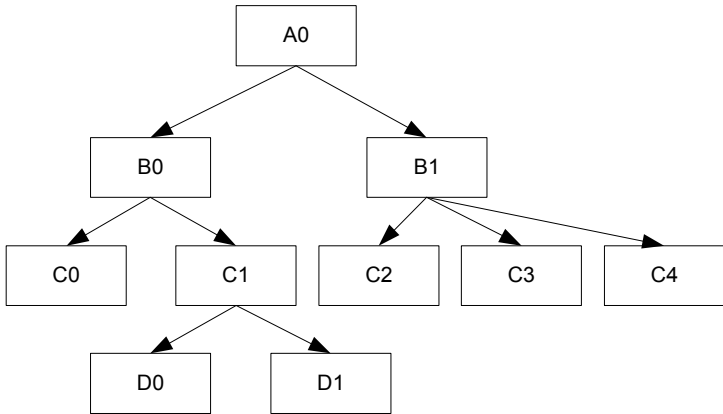
Tačiau dažniausiai duomenų bazių valdymo sistemos leidžia išreikšti tik du pirmuosius kardinalumo apribojimo tipus. Todėl projektuojant loginį modelį reikia atsižvelgti į šiuos apribojimus.

Bet kurią duomenų struktūrą galima vaizduoti tiek rodyklių, tiek lentelių forma.

3.2.2. Hierarchinis duomenų modelis

Hierarchinės duomenų bazės – tai tokios duomenų bazės, kuriose dalykinės srities esybės sudaro hierarchinę struktūrą (pvz., medis).

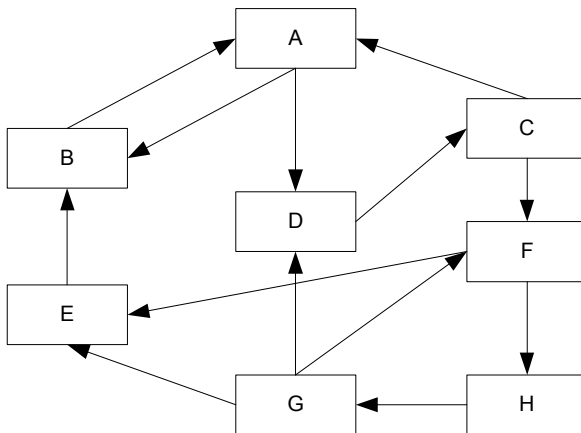
Pagrindinis hierarchinių duomenų bazių trūkumas – ribota prieiga prie saugomų esybių (pvz., norint rasti esybę D1, reikia žinoti A0, B0 ir C1) reikšmes.



3.2 pav. Hierarchinės duomenų bazės modelis, išreiškiamas medžio tipo struktūra

3.2.3. Tinklinis duomenų modelis

Tinklinės duomenų bazės – tai tokios duomenų bazės, kuriose dalykinės srities esybės sudaro tinklinę duomenų struktūrą (pvz., tokią struktūrą gali išreikšti orientuotas grafas, kai rodyklės turi kryptį, arba tinklas).



3.3 pav. Tinklinės duomenų bazės pavyzdys. Duomenų bazė sudaryta pasitelkiant kryptinį grafa

Pagrindinis tinklinių duomenų bazių trūkumas – ribota prieiga prie saugomų esybių. Pranašumas – galima sudaryti tokius ryšius, kurie geriausiai išreikštų esybių paieškos tikimybę.

3.3. RELIACINIŲ DUOMENŲ BAZIŲ PROJEKTAVIMAS

Duomenų bazės projektavimo rezultatas yra loginė duomenų bazės struktūra. Sprendžiant šį uždavinį ieškoma kuo geresnio rezultato, kad duomenų bazė turėtų kiek galima daugiau gerų savybių. Duomenų bazės struktūros kokybei įvertinti gali būti taikomi formalūs metodai, padedantys kuo optimaliau suprojektuoti duomenų bazės struktūrą.

Reliaciniame modelyje duomenys pateikiami lentelėmis. Matematikoje lentelę atitinka santykis (angl. *relation*). Lentelės stulpeliai gali būti vadinami atributais arba parametrais. Duomenų bazių terminologijoje jie vadinami laukais. Stulpelis turi turėti prasminį pavadinimą – tai atributo pavadinimas. Visų leistinų stulpelio reikšmių aibė vadinama domenu pavadinimas.

Reliacinio modelio kūrimo procesas susideda iš keleto žingsnių, kuriuos labai svarbu taisyklingai atlikti:

- ✚ Identifikuoti visus *atributus*, kurie bus naudojami kuriamoje duomenų bazėje.
- ✚ Identifikuoti atitinkamus *esybių tipus*, kuriems bus sudaromos lentelės su tam tikrais atributais (išskirtais laukais), taikant taisykles, leisiančias minimizuoti duomenų dubliavimą ir išvengti pertekliaus (šiam tikslui taikomos priemonės vadinamos normalizacija).
- ✚ Identifikuoti lentelių ryšius, kurie pagal prasmę sujungs atributus, išskirtus reikiamuose esybių tipuose. Tai atliekama pažymint, kuris iš kiekvienos lentelės atributų yra jungiamasis elementas. Kiekvienas jungiamasis atributas vaizduos vieną ryšio „vienas su daugeliu“ galą.

Pateiktieji atributai, esybių tipai ir ryšiai sudarys panašaus pobūdžio kaip esybių modelis reliacinio modelio diagramą. Šiuos du

modelius būtų galima palyginti ir pasirinkti tokį, kuris atspindėtų geriausias savybes.

Norint nustatyti geriausią duomenų modelio, kuris tinkamiausiai atspindėtų sistemos veiklos reikalavimus, struktūrą, įvertinama esybių tipų pasirinkimo kokybė, nustatomi optimalūs išskiriamų laukų ilgiai, įvertinami kiekvienos lentelės įrašų kiekiai, taigi projektas yra tobulinamas, modelis koreguojamas.

Duomenų bazė turėtų būti projektuojama laikantis duomenų vientisumo principų, optimaliai išnaudojant užimamą kompiuterinę atmintį, vengiant duomenų pertekliaus, duomenų atnaujinimo anomalijų ir kitų problemų. Aptarsime pagrindines problemas, pasitaikančias projektuojant duomenų bases.

- ✚ Duomenų perteklius – tai duomenų pasikartojimas, dubliavimas duomenų bazėje.
- ✚ Duomenų vientisumas – duomenų suderinimas (tikslumas ir neprieštaringumas) duomenų bazėje.
- ✚ Atnaujinimo anomalija – duomenų tarpusavio prieštaravimas dėl duomenų pertekliaus ir dalinio atnaujinimo.
- ✚ Pašalinimo anomalija – nenumatytas duomenų praradimas dėl kitų duomenų šalinimo.
- ✚ Įvedimo anomalija – kai neįmanoma įvesti duomenų dėl kitų duomenų nebuvimo.

Duomenų vientisumo sąlygos – tai vartojamų sąvokų (kategorijų), nuorodų vientisumo, funkcinių ryšių ir pan. reikalavimai.

Reliacinėje lentelėje eilutės dažniausiai yra konkrečių realaus pasaulio objektų (esybių, asociacijos ryšių) atvaizdai. Duomenų vientisumo reikalavimai siejasi su lentelės raktų sąvoka ir viena-reikšmiška objektų identifikacija. Norėdami duomenų bazėje rasti duomenų apie konkretų objektą (egzempliorių), turime žinoti raktų reikšmes. Objektas neturi reikšmės duomenų bazėje, jei bent viena raktų atributo reikšmė yra nežinoma. Todėl pagal sąvokų (kategorijų) vientisumo reikalavimą joks lentelės raktų atributas nė vienoje eilutėje negali turėti tuščios (*NULL*) reikšmės.

Kitame skyriuje detaliau aptarsime, kaip suprantamas duomenų bazės raktas.

Nuorodų vientisumo reikalavimas gali būti formuluojamas taip: kiekvieno išorinio rakto reikšmė duomenų bazėje turi sutapti su viena pirminio rakto reikšme lentelėje, kurią nurodo išorinis raktas.

3.3.1. Duomenų bazės rakto samprata

Kad lentelėje saugomus duomenis būtų galima vienareikšmiškai identifikuoti, įvedama *rakto sąvoka*.

Raktas – tai minimali atributų aibė, vienareikšmiškai apibrėžianti kiekvieno atributo reikšmę eilutėje (t. y. konkretų lentelės įrašą). Lentelės L raktu vadinamas toks jos atributų aibės poaibis R , kurio:

- + Rakto atributų reikšmės vienareikšmiškai apibrėžia visų lentelės L atributų reikšmes, t. y. dviejų skirtingų lentelės eilučių visų rakto aibės R atributų reikšmės negali sutapti (vienareikšmiška identifikacija).
- + Nė vienas rakto atributų aibės R poaibis neturi turėti vienareikšmiškos identifikacijos (pertekliaus nebuvimas).

Norėdami geriau suprasti rakto funkciją duomenų bazėje, pagrafinėkime *Asmenų* lentelę. Šioje lentelėje saugomi duomenys atitinka realaus pasaulio asmenų klasės egzempliorius. Būna tokių atvejų, kai du skirtingi asmenys turi tą pačią pavardę ir vardą, pavyzdžiui, Jonaitis Jonas šioje lentelėje gali kartotis. Tačiau duomenų bazėje turime atskirti šiuos du skirtingus asmenis (t. y. jų įrašus lentelėje *Asmenys*). Siekiant vienareikšmiškai identifikuoti reikiamą realybės vienetą, projektuojant duomenų bazę reikia turėti raktą, kuris vienareikšmiškai nusakytų konkretų realybės objektą. Tokiu atveju skirtingi tikrovės objektai, turintys tą patį pavadinimą, bus atskiriami ir duomenų bazėje. Duomenų lentelės raktas gali būti vienas atributas arba kelių atributų derinys.

Renkantis raktą, kuris vienareikšmiškai identifikuotų įrašus *Asmenų* lentelėje, galimi keli būdai:

- + Raktas gali būti kelių laukų (parametrų), pavyzdžiui, trijų laukų derinys $\{Vardas, Pavardė, Gimimo\ data\}$;
- + Raktas gali būti vienas laukas, tačiau tada jis turi vienareikšmiškai nusakyti kiekvieną esybę, pavyzdžiui, pa-

pildomai įvestas laukas, nurodantis asmens tabelio numerį, asmens kodą, arba tiesiog identifikatorius (*ID*). Konkreti šio lauko reikšmė vienareikšmiškai nurodys reikiamą egzempliorių.

Laukas *ID* gali būti nurodomas kaip *Asmenų* lentelės raktas.

Pasirinkti raktą, sudarytą iš keleto atributų, yra gana sudėtinga ir kai kuriose duomenų bazių valdymo sistemose tai nėra atliekama. Todėl lengviau nurodyti vieną atributą ir jį naudoti kaip raktą, kuris vienareikšmiškai identifikuoatų reikiamą lentelės įrašą.

Raktas, kurį sudaro daugiau nei vienas atributas, vadinamas *sudėtiniu raktu*.

Lentelėje gali būti keli atributų rinkiniai, turintys rakto savybę. Jie vadinami *galimais raktais*. Raktas, kuriuo lengviau ir paprasčiau naudotis, pažymimas *pirminiu raktu*. Kiti raktai yra antriniai.

3.3.2. Duomenų bazės struktūros normalizacija

Norint taisyklingai suprojektuoti reliacinę duomenų bazę, reikia laikytis normalizacijos principų.

Normalizacija – tai procesas, leidžiantis laikantis tam tikrų taisyklių išskaidyti duomenis į lenteles, siekiant supaprastinti ryšius, išvengti duomenų dubliavimosi, išlaikyti duomenų vientisumą. Normalizacija leidžia reliacinėms lentelėms suteikti standartinį pavidalą, išvengiant nesklandumų, kurių atsiranda dėl neteisingo duomenų bazės projektavimo.

Lentelių skaidymu vadinamas lentelės padalijimas į kelias lenteles, siekiant išvengti anomalijų ir neprarasti duomenų vientisumo.

Taigi panagrinėkime duomenų normalizacijos procesą. Dažniausiai duomenų bazės lentelių struktūra projektuojama išskiriant pirmą (1 NF), antrą (2 NF), trečią (3 NF) ir ketvirtą (4 NF) normalines formas.

3.3.3. Pirma normalinė forma

Pirma normalinė forma – tai tokia duomenų išskaidymo į atskiras lenteles forma, kai visos atributų reikšmės yra atomai.


Atomu laikoma reikšmė, kuri nėra nei aibė, nei sąrašas, t. y. atomo reikšmės nesudaro tokių junginių, kurie gali būti išreiškiami aibėmis, pasikartojančiomis grupėmis ar sąrašais.

Kitais žodžiais tariant, pirma normalinė forma – tai toks lentelių skaidymo būdas, kai konkretus kiekvieno lauko duomuo (reikšmė) nesudaro aibės.

Pirmos normalinės formos (1 NF) reikalavimas duomenų bazės lentelėms yra per silpnas, kad lentelių schemą būtų galima laikyti suprojektuota pakankamai gerai. Todėl nagrinėjamos ir aukštesnės eilės normalinės formos. Kuo aukštesnė normalinė forma, tuo didesni reikalavimai keliami duomenų bazės schemai.

Aukštesnėms normalinėms formoms apibrėžti naudojama funkcinio ryšio sąvoka. Jo esmė – vieno atributo reikšmės eilutėje turi vienareikšmiškai apibrėžti kitų atributų reikšmes.

Projektuojant duomenų bazę pirmiausia identifikuojami atributai. Nustatant išskiriamų parametrų sąrašą taikoma keletas standartizuotų veiksmų.

Pardavimo važtaraštis		 <p>skaitmeninė technika</p>	
Važtaraščio Nr.	_____		
Pirkėjo Nr.	_____		
Pirkėjo vardas	_____		Pirkėjo adresas
Pirkėjo pavardė	_____		_____
Važtaraščio data	_____		_____
Prekės Nr.	Prekės pavadinimas	Kiekis	

3.4 pav. Važtaraščio pavyzdys

Projektuojant duomenų struktūros pirmą normalinę formą (1 NF) taikoma taisyklė, įvedanti esybių tipų sąrašo sudarymo apribojimus, t. y. esybės tipui (lentelei) negalima priskirti tokių atributų, dėl kurių esybės (sinonimai – objektas, įrašas) toje pačioje lentelėje kartotųsi.

Panagrinėkime konkretų pavyzdį – važtaraščių apskaitos duomenų bazės schemas sudarymą.

Pavyzdžiui, esybės tipo *Pardavimo važtaraštis* visi atributai, susiję su konkrečių prekių pardavimu, gali pasireikšti keletą kartų. Suprantama, kad pirkėjas, kurio važtaraštis pildomas, gali pirkti daug skirtingos rūšies prekių.

Funkcinio ryšio terminais, jeigu atributas keletą kartų kartojasi toje pačioje esybės lentelėje, negalima nurodyti, kad bus rasta tik viena kiekvieno rakto reikšmė. Dėl šios priežasties kartojimąsi lemiančius parametrus skaidome į lenteles.

Dokumentas <i>Pardavimo važtaraštis</i>			
Nenormalizuota forma	Pirma normalinė forma	Antra normalinė forma	Trečia normalinė forma
Važtaraščio Nr. Pirkėjo Nr. Važtaraščio data Pirkėjo vardas Pirkėjo pavardė Pirkėjo adresas Prekės Nr. Prekės pavadinimas Prekių kiekis			

3.5 pav. Pardavimo važtaraščio atributų identifikacija prieš sudarant normalines formas

Taigi normalizacija reikalauja, kad sudarant pirmos normalinės formos esybių tipų sąrašą būtų pašalintos visos grupės (lauko reikšmių aibės arba sąrašai).

Pardavimo važtaraščio struktūrą projektuojant į duomenų bazės struktūrą, pirmiausiai taikoma pirmos normalinės formos sudarymo taisyklė, pagal kurią atributų sąrašas (3.6 pav.) skaidomas į dvi lenteles. Vienoje yra grafos *Važtaraščio Nr.*; *Pirkėjo Nr.*; *Važtaraščio data*; *Pirkėjo vardas*; *Pirkėjo pavardė*; *Pirkėjo adresas*; kitoje – *Važtaraščio Nr.*; *Prekės Nr.*; *Prekės pavadinimas*; *Prekių kiekis* (3.6 pav.). Lentelių pirminiai raktai pabraukti.

Dokumentas <i>Pardavimo važtaraštis</i>			
Nenormalizuota forma	Pirma normalinė forma	Antra normalinė forma	Trečia normalinė forma
<u>Važtaraščio Nr.</u> Pirkėjo Nr. Važtaraščio data Pirkėjo vardas Pirkėjo pavardė Pirkėjo adresas Prekės Nr. Prekės pavadinimas Prekių kiekis	<u>Važtaraščio Nr.</u> Pirkėjo Nr. Važtaraščio data Pirkėjo vardas Pirkėjo pavardė Pirkėjo adresas <u>Važtaraščio Nr.</u> <u>Prekės Nr.</u> Prekės pavadinimas Prekių kiekis		

3.6 pav. Pirmos normalinės formos sudarymas

3.3.4. Antra normalinė forma


Antrą ir trečią normalines formas riboja funkcinė priklausomybė. Jos rodo ryšius tarp raktinių ir neraktinių atributų.

Funkcinė priklausomybė (FP) – vieno atributo reikšmė kortelėje (stulpelyje) vienareikšmiškai apibrėžia kito atributo reikšmę.

Funkcinės priklausomybės F ryšio kairiosios dalies atributai vadinami determinantu. Determinantas – tai atributai, kurių reikšmės apibrėžia kitų atributų reikšmes. Funkcinio ryšio sąvoka yra sąvokos *lentelės visraktis* apibendrinimas.

Lentelė $L(R)$ yra *antros normalinės formos (2 NF) f sąryšių* ai-
bės F atžvilgiu, jeigu ji yra pirmos normalinės formos ir kiekvienas jos nepirminis atributas visiškai priklauso nuo kiekvieno lentelės L rakto R .

Antra normalinė forma – toks atributų sąrašo išskaidymas, kai kiekvienas neraktinis atributas funkciškai priklauso nuo viso rakto (visų raktinių atributų), t. y. negali priklausyti nuo rakto dalies (vieno iš rakto atributų).

Dokumentas <i>Pardavimo važtaraštis</i>			
Nenormalizuota forma	Pirma normalinė forma	Antra normalinė forma	Trečia normalinė forma
<u>Važtaraščio Nr.</u> Pirkėjo Nr. Važtaraščio data Pirkėjo vardas Pirkėjo pavardė Pirkėjo adresas Prekės Nr. Prekės pavadinimas Prekių kiekis	<u>Važtaraščio Nr.</u> Pirkėjo Nr. Važtaraščio data Pirkėjo vardas Pirkėjo pavardė Pirkėjo adresas <u>Važtaraščio Nr.</u> <u>Prekės Nr.</u> Prekės pavadinimas Prekių kiekis	 <u>Važtaraščio Nr.</u> <u>Prekės Nr.</u> Prekių kiekis <u>Prekės Nr.</u> Prekės pavadinimas	

3.7 pav. Antros normalinės formos sudarymas

Kad duomenų struktūra tenkintų antros normalinės formos reikalavimus, vykdomi keli skaidymo etapai:

- ✚ Sukuriama nauja lentelė, kurios atributai yra pradinės lentelės atributai, netenkinantys funkcinės priklausomybės sąlygų. Funkcinės priklausomybės determinantas tampa naujos lentelės raktu.
- ✚ Atributas, esantis dešinėje FP pusėje, išmetamas iš pradinės lentelės.
- ✚ Jeigu yra daugiau negu viena FP , pažeidžianti 2 NF, tai pirmieji du etapai kartojami esant kiekvienai papildomai FP .
- ✚ Jeigu vienas determinantas turi kelias FP , tai visi nuo jo funkciškai priklausantys atributai surašomi kaip neraktiniai atributai į lentelę, kurios raktas bus determinantas.

Važtaraščio duomenų bazės struktūros sudarymo pavyzdyje antra normalinė forma sudaroma taip, kaip pavaizduota 3.7. pav.

3.3.5. Trečia normalinė forma

Lentelė $L(R)$ yra *trečios normalinės formos* (3 NF) f sąryšių aibės F atžvilgiu, jeigu ji yra 1 NF ir nėra pirminių atributų, tranzityviai priklausančių nuo rakto.

Trečia normalizacijos taisyklė nurodo, kad ne tik kiekvienas atributas turi būti priklausomas nuo rakto, bet jis taip pat neturi funkciškai priklausyti nuo kitų sąrašo atributų.

Yra ir kitas 3 NF kriterijus, bet jis logiškai yra silpnesnis. Šis kriterijus teigia, kad lentelė tenkina 3 NF, jeigu ji neturi tranzityvinių priklausomybių.

Tranzityvinė priklausomybė atsiranda, kai neraktinis atributas funkciškai priklauso nuo vieno ar daugiau neraktinių atributų.

Trečios normalinės formos aprašymo pradininkai yra Boisas ir Koddas [5], todėl ši forma kartais literatūroje dar vadinama Boiso-Koddo normaline forma.

Taikant Boiso-Koddo kriterijų atsižvelgiama į šiuos atvejus:

- ✚ Neraktinis atributas priklauso nuo raktinio atributo, įeinančio į sudėtinį raktą (pažeidžia 2 NF kriterijų);

- ✚ Raktinis atributas, įeinantis į sudėtinį raktą, priklauso nuo neraktinio atributo. O tai reiškia, kad tenkinama ir tranzityvinės priklausomybės sąlyga.

Trečios normalinės formos sudarymo pavyzdys pateikiamas 3.8 pav.

Dokumentas <i>Pardavimo važtaraštis</i>			
Nenormalizuota forma	Pirma normalinė forma	Antra normalinė forma	Trečia normalinė forma
<u>Važtaraščio Nr.</u> Pirkėjo Nr. Važtaraščio data Pirkėjo vardas Pirkėjo pavardė Pirkėjo adresas Prekės Nr. Prekės pavadinimas Prekių kiekis	<u>Važtaraščio Nr.</u> Pirkėjo Nr. Važtaraščio data Pirkėjo vardas Pirkėjo pavardė Pirkėjo adresas	→	<u>Važtaraščio Nr.</u> <u>Pirkėjo Nr.</u> Važtaraščio data <u>Pirkėjo Nr.</u> Pirkėjo vardas Pirkėjo pavardė Pirkėjo adresas
	<u>Važtaraščio Nr.</u> <u>Prekės Nr.</u> Prekės pavadinimas Prekių kiekis	<u>Važtaraščio Nr.</u> <u>Prekės Nr.</u> Prekių kiekis	→
		<u>Prekės Nr.</u> Prekės pavadinimas	→

3.8 pav. Trečios normalinės formos sudarymas

3.3.6. Ketvirta normalinė forma

Ketvirtajai normalinei formai apibrėžti funkcinio ryšio sąvokos nepakanka. Tam įvedama funkcinių ryšių apibendrinanti daugiareikšmio ryšio sąvoka.

Sudarant ketvirtą normalinę formą, atributų išskaidymas į lenteles turi tenkinti 3 NF ir nepriklausyti daugeliui reikšmių.

Lentelės, turinčios 1 NF, reikšmės negali būti reikšmių aibė. Bet yra daug modeliavimo situacijų, reikalaujančių, kad atributo reikšmė būtų aibė. Pavyzdžiui, universiteto dėstytojas dėsto kelis dalykus ir dalyvauja keliose komisijose. Yra keletas tokio pavyzdžio realizavimo reliacinėje duomenų bazėje būdų, tai minimalaus įrašų, turinčių tuščias reikšmes, skaičiaus užtikrinimas, sąrašinių reikšmių likvidavimas.

3.4. DUOMENŲ BAZIŲ VALDYMO SISTEMOS

Duomenų bazės valdymo sistema (DBVS) vadinama programine įranga, skirta duomenų bazėms kurti, saugoti ir įvairiais būdais apdoroti.

Svarbiausios duomenų bazių valdymo sistemų funkcijos yra šios:

- ✚ Duomenų bazės struktūros projektavimas;
- ✚ Duomenų bazės pildymas, kaupimas, redagavimas;
- ✚ Naršymas ir judėjimas duomenų bazėje;
- ✚ Duomenų peržiūra, paieška, rikiavimas ir kitas tvarkymas;
- ✚ Taikomųjų vartotojo programų kūrimas;
- ✚ Ataskaitų kūrimas.

Bet kurią DBVS sudaro priemonės duomenų bazės struktūrai projektuoti, duomenims įvesti, papildyti ir modifikuoti. Šiuolaikinė DBVS suteikia vartotojui galimybes lengvai surasti ir atrinkti reikalingus duomenis pagal vieną ar kelis kriterijus, juos rikiuoti, grupuoti, vaizduoti pageidaujama forma. Tam tikrais laiko tarpais sukaupią informaciją reikia apibendrinti, atlikti jos analizę, vykdyti matematinius, statistinius skaičiavimus ir kurti įvairių formų ataskaitas. Vartotojai prireikus rašo bei derina savas taikomąsias programas, naudodamiesi tam tikromis DBVS priemonėmis. DBVS turi atlikti ir daug kitų svarbių funkcijų: duomenų apsaugą, jų tikslumo, neprieštaringumo ir išsamumo kontrolę, išsaugoti DB kopijas ir kt.

Daugelį šiuolaikinių DBVS sudaro tokios dalys:

- ✚ Duomenų tvarkymo dialoginė aplinka;

- ✚ Duomenų aprašymo ir manipuliavimo jais kalba;
- ✚ Programų generatoriai.

Dialoginė aplinka yra viena svarbiausių šiuolaikinių DBVS dalių. Ji skirta interaktyviam darbui su DBVS atliekant veiksmus su visa DB arba atskirais įrašais. Šiuolaikinės dialoginės aplinkos suteikia vartotojui galimybę pasirinkti jam priimtinausią darbo su DB stilių. Galimi keli darbo su DB būdai:

- ✚ Grafinių, vizualių priemonių panaudojimas;
- ✚ Specialios instrukcijos ir kt.

Operacijoms atlikti naudojamas užklausų mechanizmas. Užklausos rezultatas – tai arba pagal tam tikras sąlygas atrinkti duomenys, arba tam tikri pakeitimai lentelėje. Duomenų bazės užklausos formuojamos specialia struktūrizuota užklausų kalba. Reliacinėse duomenų valdymo sistemose naudojama struktūrizuota užklausų kalba *SQL (Structured Query Language)*.

Kita duomenų bazių valdymo sistemų funkcija – duomenų administravimas (duomenų apsauga nuo neteisėto panaudojimo, didelio vartotojų skaičiaus, duomenų vientisumo palaikymas).

Duomenų bazių valdymo sistemas (toliau – DBVS) sąlygiškai galima suskirstyti į tokias kategorijas:

- ✚ Lokaliosios DBVS (pvz., *MS Access, Paradox*);
- ✚ Nuotolinės kliento-serverio (angl. *client-server*) DBVS (pvz., *My SQL, Sybase, Oracle, InterBase*).

Duomenų bazės vadinamos lokaliomis, kai duomenys, jų bazių valdymo sistema ir klientams skirta darbo su duomenimis dalis yra tame pačiame kompiuteryje. Paprastai tokią duomenų bazę sudaro vienas failas (pvz., *MS Access* valdo failus, kurių plėtinys yra **.mdb*). Tokios architektūros duomenų bazės netinka išskirstytosioms informacinėms sistemoms realizuoti.

Grafiniai instrumentai pagal savo veikimo principą skirstomi į projektavimo, dizaino, žynio-patarėjo ir meniu funkcijas atliekančias sistemas. Projektavimo sistemos suteikia galimybę vartotojui pačiam laisvai pasirinkti, ką ir kaip projektuoti. Meistrai ir meniu sistemos leidžia iš pateikiamų galimų veiksmų sąrašų (meniu) išsirinkti reikiamas komandas, o vartotojui telieka patvirtinti savo pasirinkimą iš daugelio siūlomų veiksmų sekų. Be minėtųjų instrumentų, dažnai

naudojami konstruktoriai, padedantys rašyti sudėtingas duomenų apskaitos ir lyginimo išraiškas, formules ir pan.




Kita svarbi DBVS dalis yra duomenų aprašymo ir manipuliavimo jais kalba. Ji skirta kurti taikomas programas, t. y. tokias programas, kurias vartotojas ar jų grupė naudoja konkrečioms, specifiniams uždaviniams spręsti. Į tokios kalbos sudėtį įeina duomenų aprašymo ir manipuliavimo jais instrukcijos (komandos). Aprašant duomenis nurodoma DB lentelių struktūra, raktiniai laukai, atskirų lentelių ryšiai. Manipuliavimo duomenimis instrukcijos – tai daugybė įvairios paskirties komandų, skirtų tiesiogiai apdoroti duomenų elementus (laukų reikšmes), lentelių įrašus, lenteles ar visą DB.

Iš daugybės manipuliavimo duomenimis instrukcijų į savarakišką grupę išskiriamos vadinamosios užklausos. Jos aprašomos ir vykdomos atitinkama kalba. Viena plačiausiai naudojamų užklausų kalbų yra *SQL (Structured Query Language* – struktūrizuota užklausų kalba). Tai gana paprasta kalba, turinti aiškias standartizuotas instrukcijas. Ši kalba įtraukta į daugumą DBVS.

Modernios DBVS turi programų generatorius. Tai specialūs instrumentai, skirti automatizuoti programų rašymą, palengvinti vartotojų, ypač programuotojų, darbą kuriant taikomas programas. Generatoriai įgalina žymiai sumažinti darbo sąnaudas, kai reikia programuoti daug laiko reikalaujančias ir dažnai pasikartojančias, pavyzdžiui, ekrano apipavidalinimo, veiksmų (menu) juostų, atskaitų sudarymo ir kitas operacijas.

3.5. PAGRINDINIAI DUOMENŲ BAZĖS KŪRIMO IR REDAGAVIMO ETAPAI

Pagrindiniai darbo su duomenų baze etapai yra:

-  Duomenų bazės struktūros projektavimas;
-  Duomenų bazės užpildymas duomenimis;
-  Duomenų bazės struktūros redagavimas ir duomenų atnaujinimas.

Naujos DB kūrimas prasideda nuo bazės struktūros, t. y. bazės lentelių struktūros projektavimo. Formuojant konkrečios lentelės struktūrą, nurodomi:

- ✚ Lentelės laukų pavadinimai;
- ✚ Laukų duomenų tipai;
- ✚ Laukų ilgiai;
- ✚ Duomenų formatai;
- ✚ Priskiriamos rakto funkcijos, indeksuojami laukai.

Papildomai gali būti nurodyti ir kiti dalykai, pavyzdžiui, duomenų laukų formatai, nutylėtos laukų reikšmės, t. y. reikšmės, kuriomis laukas užpildomas automatiškai. Vykdomos dalinės įvedamų duomenų kontrolės funkcijos. Suformavus lentelių struktūrą, nustatomi raktiniai ir lentelės siejantys laukai, nurodomi ryšių, reikalingų lentelėms susieti į vientisą duomenų bazę, tipai.

Baigus kurti DB struktūrą, bazę užpildoma konkrečiais vartotojo duomenimis. Vėliau duomenų bazę tenka redaguoti (įterpti naujus įrašus, pasenusius duomenų elementus pakeisti naujais ir pan.). DBVS leidžia redaguoti ir DB struktūrą, ir jos turinį, t. y. duomenų elementus.

Redaguojant DB struktūrą, daugiausia tenka:

- ✚ Keisti lauko duomenų tipą, dydį;
- ✚ Įterpti papildomą lauką;
- ✚ Šalinti nebereikalingą lauką.

Čia pažymėtina, jog geriau iš karto detaliai apgalvoti ir suprojektuoti efektyvią DB struktūrą, kad vėliau netektų jos keisti. Žymiai dažniau reikia redaguoti DB turinį, o ne bazės struktūrą. Svarbiausios DB turinio redagavimo operacijos yra šios:

- ✚ Naujo įrašo įterpimas;
- ✚ Atskiro lauko ar viso įrašo užpildymas konkrečiais duomenimis (reikšmėmis);
- ✚ Įrašo laukų reikšmių modifikavimas;
- ✚ Įrašo pašalinimas.

Šiomis operacijomis nesunku konstruoti kitas, sudėtingesnes operacijas. Pavyzdžiui, galima automatiškai pakeisti tam tikro lauko reikšmės visoje lentelėje, išvalyti lauką ar net visą lentelę.

DB turiniui redaguoti labai plačiai naudojamos vadinamosios formos.

Forma – tai tam tikras langas, skirtas vaizduoti ir redaguoti laukų reikšmes iš vienos ar kelių DB lentelių. Langas atitinkamai apipavidalinamas, laukų reikšmės jame išdėstomos laisva tvarka ir

vaizduojamos vartotojui patogiu formatu. Formoje turi būti ir duomenų vaizdavimo valdymo įrankiai, dar kitaip vadinami naršymo įrankiais.

Galimos konkretaus užsakymo duomenų turinio redagavimo formos pavyzdys pateiktas 3.9 paveiksle.

<i>Užsakymo numeris</i>	0331	<i>Data</i>	2004 01 26
<i>Pirkėjas</i>	Jankauskienė		
<i>Prekė</i>	Kompiuteris		
<i>Kiekis</i>	10		
<i>Kaina</i>	5000,00		
	Kitas	Ankstesnis	Išėjimas

Aiškinamasis tekstas ↑

↑ *Valdymo mygtukai*

↑ *Lauko reikšmė*

3.9 pav. Duomenų įvedimo formos pavyzdys

Įrankiai pateikiami kaip tam tikri mygtukai, skirti atitinkamiems nurodymams atlikti, „pereiti prie tolesnio įrašo“, „redaguoti įrašą“ ir pan. Formos naudojamos ne tik DB redaguoti, bet ir įvesti naujiems duomenims, peržiūrėti, surasti reikiamą įrašą lentelėje.

3.6. DUOMENŲ PAIEŠKOS PRIEMONĖS

Paiška yra manipuliavimo duomenimis operacija ar operacijų seka, kurios paskirtis – iš vienos arba kelių DB lentelių surasti konkrečius duomenis. Paieška yra viena svarbiausių manipuliavimo duomenimis operacijų. Be šios operacijos neįmanoma atlikti daugelio kitų operacijų (įrašų modifikavimo, įterpimo, pašalinimo). Vartotojui labai aktualu paiešką atlikti greitai, patogiai, lanksčiai. Galima teigti, kad paieškos greitis nulemia viso darbo su DB efektyvumą.

Paiešką duomenų bazėje galima organizuoti:

- ✚ Naudojantis grafiniais, vizualiaisiais instrumentais;
- ✚ Rašant instrukcijas specialia kalba.

Užklauskos duomenų bazėse atlieka informacines paslaugas. Jos skirtos informacijos paieškai atlikti.

Microsoft Access turi kelis užklauskų tipus:

- ✚ Atrankos užklauskos, vykdančios duomenų paiešką pagal atrankos kriterijus (*Select Query*), skirtos ieškoti įrašų ir rezultatą pateikti lentelės forma;
- ✚ Apibendrintos užklauskos. Jos gali būti vadinamos suvestinėmis, kryžminėmis suvestinėmis užklausomis (*Crosstab Query*), agregatinėmis užklausomis. Tokio tipo užklauskos pateikia statistinių skaičiavimų (sumos, kiekio, vidurkio) rezultatus, gautus pagal lentelės laukų duomenis, skaičiuojant juos pagal reikiamus požymius;
- ✚ Duomenis keičiančios užklauskos. Jos skirtos duomenų įrašams arba struktūrai keisti. Pavyzdžiui, duomenims trinti, naujoms lentelėms kurti, joms keisti (*Action Query*) ir pan.:
 - Delete Queries* – pašalina iš lentelės nereikalingus įrašus;
 - Update Queries* – pakeičia reikiamus įrašus;
 - Append Queries* – lentelės gale prideda naujus įrašus, neatitinkančius reikalaujamų kriterijų;
 - Make-Table Queries* – iš atrinktų įrašų sukuria naujas lenteles;
- ✚ Užklauskos įvedant parametrus (*Parameter Query*) leidžia įvesti atrankos sąlygas. Šiuo atveju užklausa prašo įvesti ieškomo parametro reikšmę ir tuomet atrenka įrašus, kurių šie parametrai sutampa.

Su duomenų paieškos, užklauskų bei ataskaitų kūrimo priemonėmis plačiau susipažinsime praktiniuose darbuose.

3.7. DUOMENŲ BAZIŲ PRAKTINIAI DARBAI IR METODINIAI NURODYMAI

3.7.1. Konferencijos organizavimo duomenų bazės kūrimas

1. Darbo tikslas

Išmokti teisingai sudaryti duomenų bazės loginę struktūrą, duomenų saugojimo lenteles ir ryšius tarp lentelių. Sukurti duomenų įvedimo formas. Susipažinti su duomenų bazių valdymo sistemos (DBVS) *Microsoft Access* darbo aplinka.

2. Darbo užduotys

1. Sukurkite naujos duomenų bazės loginę struktūrą, skirtą konferencijos dalyviams registruoti.

2. Iškviškite programą *MS Access*. Pasirinkite naujos duomenų bazės kūrimo režimą.

3. Programai pasiūlius, išsaugokite savo naujai kuriamą duomenų bazę diske, pavyzdžiui, C, D, savo sukurtame kataloge. Duomenų bazei suteikite prasminį pavadinimą.

Visos kuriamos lentelės bus saugomos vienoje bendroje duomenų bazėje, kuri saugant įvardijama prasminiu semantiniu pavadinimu, pavyzdžiui, *Konferencija.mdb*.

4. Sukurkite naują duomenų saugojimo lentelių loginę struktūrą, skirtą konferencijos dalyviams registruoti. Pavadininkite lenteles *Autoriai*, *Organizacijos* ar pan. Išsiaiškinkite galimus laukų duomenų tipus, jų panaudojimo prasmę ir formatus. Nurodykite prasminius duomenų laukų pavadinimus, pasirinkite duomenų formato tipus ir laukų ilgius.

5. Įveskite duomenis į lentelę *Autoriai*.

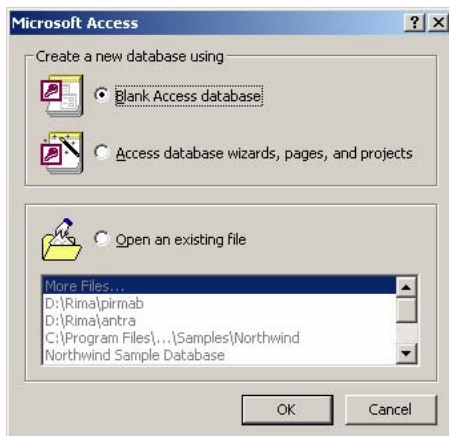
6. Sukurkite naują duomenų saugojimo lentelę, skirtą straipsniams ir recenzentų duomenims įvesti. Suteikite lentelėms semantinius pavadinimus, pavyzdžiui, *Straipsniai*, *Recenzentai*. Nustatykite duomenų tipus ir laukų ilgius.

7. Įveskite duomenis apie konferencijai pateiktus straipsnius ir recenzentus ir panašiai bei sukurkite šias esybes siejančias lenteles.

3. Metodiniai nurodymai

Kaip iškviešti programą Microsoft Access ir išsaugoti duomenų bazės failą?

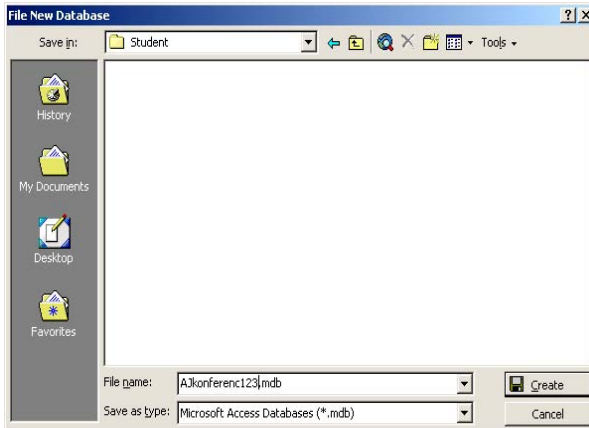
Iškviečiant programą *Access* (*Start*→*Programms*→*Microsoft Access*), programa siūlo pasirinkti iš sąrašo ir atverti sukurtą duomenų bazę arba kurti naują (3.10 pav.). Norint atverti jau sukurtą duomenų bazę (*Open an existing file*), ji yra pasirenkama iš sąrašo. Jeigu sukurtosios duomenų bazės pavadinimo nėra, pasirenkama komanda *More files*. Norint kurti naują duomenų bazę, pasirenkamas *Blank Access database* veiksmas.



3.10 pav. Duomenų bazės kūrimo langas su pasirinktais parametrais naujai duomenų bazei kurti

Pradedant kurti naują duomenų bazę, programa pasiūlo ją išsaugoti. Šį duomenų bazės failą išsaugokite kietajame diske, pavadindami, pavyzdžiui, *konferencija.mdb*.

Šiame DB faile bus saugomi visi sukurti duomenų bazės elementai: lentelės, formos, užklauskos, ataskaitos, jų pakeitimai ir papildymai.



3.11 pav. Duomenų bazės pavadinimo ir išsaugojimo vietos parinkimo langas

Visi veiksmai atliekami taip, kaip ir vykdant komandą *File*→*Save As*. Pasirenkamas diskas, aplankas, duomenų bazės failui suteikiamas pavadinimas ir spaudžiamas *Create* mygtukas.

Duomenų bazės sandaros elementai

Atitinkamu pavadinimu išsaugojus naują duomenų bazę programa atveria darbo langą, kuriame pateikiami duomenų bazės sudedamųjų dalių kūrimo objektai (*Objects*). Jie yra pagrindiniai išvardyti duomenų bazės sandaros elementai, kuriais galima kurti:

- ✚ ***Lentelės (Tables)***, jose laikomi pagal duomenų apdorojimo poreikius sugrupuoti duomenys;
- ✚ ***Formos (Forms)***, jos leidžia pasirinkti patogią formą duomenims įvesti, redaguoti ir peržiūrėti;
- ✚ ***Užklausas (Queries)***, jose aprašomi ieškomų duomenų atrankos kriterijai bei atrinkamų duomenų grupavimo būdai;
- ✚ ***Ataskaitas (Reports)***, remiantis jomis galima pateikti iš bazės išrinktus duomenis, skaičiavimų rezultatus, iliustracijas, suteikiančias duomenims tam tikrą apipavidalinimą. Jos gali būti pateiktos ekrane ir spausdinamos;

✚ **Makrokomandas** (*Macros*), jomis galima automatizuoti dažnai kartojamų komandų grupių vykdymą;

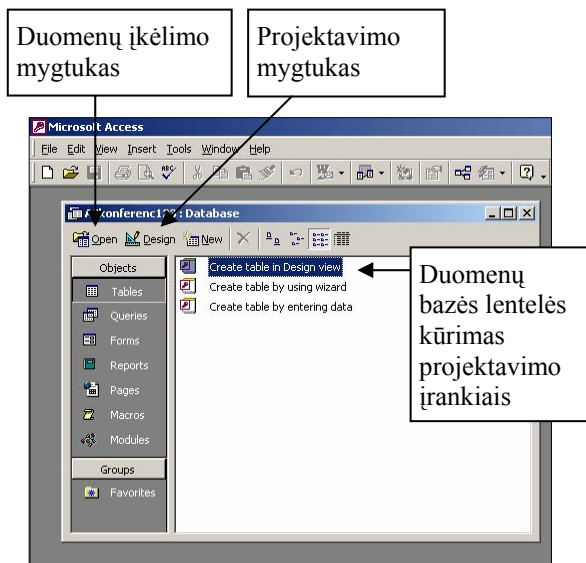
✚ **Modulius** (*Modules*), jų dėka atliekami duomenų apdorojimo aprašymai DBVS programavimo kalba *Visual Basic*.

Pradedant kurti duomenų bazę šių objektų sąrašai būna tušti.

Didelės apimties duomenų bazėse įvairių tipų objektai gali būti jungiami į grupes, sukurtųjų grupių valdymo mygtukai yra grupės srityje (*Groups*). Jeigu grupių nėra, šioje srityje būna tuščia grupė *Favorites*.

Kaip sudaryti lentelės projektą, pasirinkti duomenų tipus ir nustatyti laukų ilgus?

Duomenų bazę pradedama kurti loginio modelio projektu (3.3. skyrius). Suprojektuosime konferencijos dalyvių duomenų lentelę įvesdami laukų pavadinimus, jų tipus ir laukų ilgius. Naujai duomenų lentelei kurti pasirenkamas objektas *Tables* ir dukart spragtelint kai-riuoju pelės klavišu pasirenkamas darbo režimas *Create table in Design view*. Pradėti kurti lentelę galima ir kitu būdu, t. y. pasirinkus objektą *Tables* spaudžiamas mygtukas *New*, pasirodžiusiame dialogo lange *New Table* spaudžiamas ir pasirenkamas *Design view* režimas.



3.12 pav. Lentelės kūrimo įrankių pasirinkimas

Stulpelyje *Field Name* įvedame laukų pavadinimus. Stulpelyje *Date Type* svarbu tinkamai nustatyti įvedamų duomenų tipą. *Field Size* dalyje nustatomas maksimalus lauko dydis.

Yra dešimt galimų duomenų tipų: teksto (*Text*), didelės atminties aprašymo (*Memo*), skaitmeninės (*Number*), automatinio skaitiklio (*AutoNumber*), loginio (*Yes/No*), pinigų (*Currency*), datos ir laiko (*Date/Time*), įterpiamų objektų (*OLE object*), hipernuorodų (*Hyperlink*).

3.1 lentelė. Laukų tipai

Duomenų tipas	Savybės ir reikšmė
Teksto (<i>Text</i>)	Simbolių rinkiniai. Didžiausias teksto lauko ilgis yra 255 simboliai.
Aprašymo (<i>Memo</i>)	Ilgı simbolių rinkiniai, kurių simbolių skaičius viršija 255. Lauko ilgis gali būti iki 65 535 simbolių. Šie laukai negali būti raktiniai ir neindeksuojami. Jie lentelėse nėra saugomi, jose paprastai yra nuorodos į šiuos laukus, todėl lentelėse greičiau vyksta duomenų rūšiavimas ir paieška.
Skaitmeninis (<i>Number</i>)	Skaitmeniniai duomenys – tai įvairaus formato skaičiai, kurie gali būti naudojami atliekant matematinius skaičiavimus. Šio tipo duomenys gali būti 1, 2, 4, 8, ar 16 baitų ilgio. Tai priklauso nuo lauko dydžio (<i>Field Size</i>). <i>Long Integer</i> dydis yra 4 baitai ir tai yra sveikųjų skaičių atkarpa nuo –2147483648 iki 2147483648.
Automatinio skaitiklio (<i>AutoNumber</i>)	Automatiškai formuojamos įrašams žymėti skirtos skaitinės reikšmės, jų laukas yra 4 baitų ilgio. Programa automatiškai formuoja šį lauką, padidindama jį vienetu arba atsitiktiniu būdu. Maksimalus įrašų skaičius lentelėje esant šiam laukui turi neviršyti 2 milijardų.
Loginis (<i>Yes/No</i>)	<i>Taip/ne</i> tipo laukas arba <i>įjungta/išjungta</i> . Programa <i>Access</i> naudoja reikšmę – 1, esant <i>taip</i> tipo laukui, reikšmę 0, esant <i>ne</i> tipo laukui. Šis laukas negali būti raktas, tačiau jį galima indeksuoti.
Pinigų (<i>Currency</i>)	Fiksuoto kablelio skaičiai, įvairių valiutų piniginiams vienetams išreikšti. 15 ženklų skiriama sveikajai daliai ir 4 – trupmeninei. Gali būti skaičiavimų argumentai.
Datos ir laiko (<i>Date/Time</i>)	Galima įvesti datą nuo 100 iki 9999 metų. Lauko ilgis – 8 baitai. <i>Access</i> suteikia galimybę rinktis įvairius datos ir laiko formatus.
Įterpiamų objektų (<i>OLE object</i>)	Iliustracijoms, vaizdo įrašams, garsui, dokumentams saugoti skirtas lauko duomenų tipas.
Hipernuorodos (<i>Hyperlink</i>)	Interneto adresai arba nuorodos su diskuose laikomais failais. Hipernuorodos užrašymo ilgis – iki 2048 simbolių. Šis laukas negali būti raktinis ar indeksuojamas.

3.2 lentelė. Skaitmeninių laukų atmainos

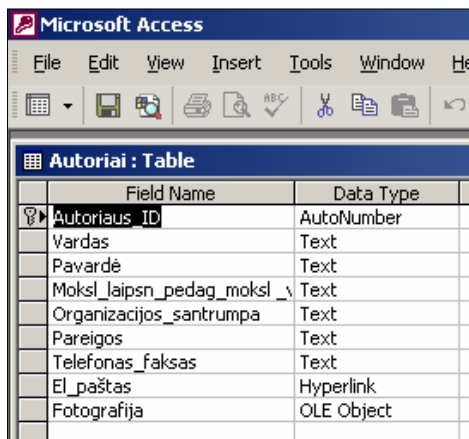
Atmaina	Dydis	Savybės
Baitas (<i>Byte</i>)	1 baitas	Sveikųjų skaičių atkarpa nuo 0 iki 255
Sveikasis (<i>Integer</i>)	2 baitai	Sveikųjų skaičių atkarpa nuo –32768 iki +32767
Dideli sveikieji skaičiai (<i>Long Integer</i>)	4 baitai	Sveikųjų skaičių atkarpa nuo –2147483648 iki +2147483647
Dešimtainiai (<i>Decimal</i>)	12 baitų	28 ženklų tikslumo sveikieji skaičiai
Realusis (<i>Single</i>)	4 baitai	Slankiojo kablelio skaičiai nuo –3,402823E38 iki +3,402823E38
Dvigubo tikslumo (<i>Double</i>)	8 baitai	Slankiojo kablelio skaičiai nuo –1,79769313486231E308 iki +1,79769313486231E308
Sisteminės žymos (<i>Replication ID</i>)	16 baitų	Unikalių sisteminių nuorodų reikšmės

3.3 lentelėje pateiktas *Autorių* lentelės projektas, siūlomi laukų pavadinimai, laukų tipai ir jų ilgiai bei jų struktūros kūrimo pavyzdys (3.13. pav.). Sukūrus lentelės struktūrą pasirenkamas raktinis laukas (pvz., *Autoriaus ID*). Šis laukas turi būti indeksuotas ir nedubliuojamas.

Jeigu įvedant duomenis į lentelę lauko ilgis pasirodys per trumpas ir duomuo, pavyzdžiui, vardas, pavardė ar panašiai, netilps, tuomet lauko ilgį darbo metu galėsite pakeisti, koreguodami loginę *DB* projekto struktūrą. Šiuo atveju kiti įvesti duomenys pildant lentelę ar formą nedings.

3.3 lentelė. Lentelės *Autoriai* projektas

Lauko pavadinimas	Lauko tipas	Lauko ilgis
Autoriaus ID	<i>AutoNumber</i>	
Vardas	<i>Text</i>	15
Pavardė	<i>Text</i>	20
Mokslo laipsn pedagog moksl vard	<i>Text</i>	20
Organizacijos santrumpa	<i>Text</i>	5
Pareigos	<i>Text</i>	30
Telefonas faksas	<i>Text</i>	30
El paštas	<i>Hyperlink</i>	25
Fotografija	<i>OLE</i>	



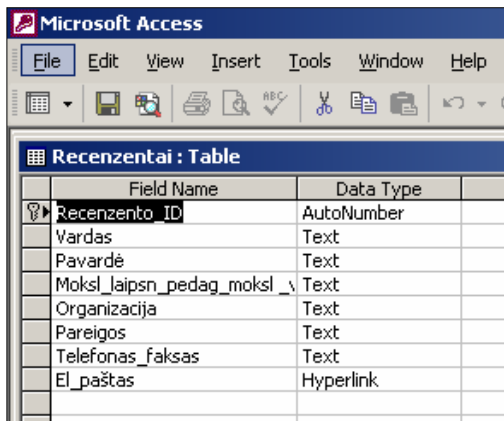
Field Name	Data Type
Autoriaus_ID	AutoNumber
Vardas	Text
Pavardė	Text
Moksl_laipsn_pedag_moksl_	Text
Organizacijos_santrumpa	Text
Pareigos	Text
Telefonas_faksas	Text
El_paštas	Hyperlink
Fotografija	OLE Object

3.13 pav. Lentelės *Autoriai* laukai ir laukų tipai

3.4 lentelėje pateikiamas *Recenzentų* lentelės projektas, siūlomi laukų pavadinimai, laukų tipai ir jų ilgiai. Jeigu lauko ilgis įvedant duomenis į lentelę pasirodo trumpas ir duomuo, pavyzdžiui, vardas, pavardė ar pan., netelpa tuomet lauko ilgį darbo metu galima pakeisti. Analogiškai sukuriamos kitos lentelės: *Straipsniai*, *Organizacijos* ir pan.

3.4 lentelė. Lentelės *Recenzantai* struktūra

Lauko pavadinimas	Lauko tipas	Lauko ilgis
Recenzento ID	<i>AutoNumber</i>	
Vardas	<i>Text</i>	15
Pavardė	<i>Text</i>	20
Moksl_laipsn_ pedag_moksl_vard	<i>Text</i>	20
Organizacijos_santrumpa	<i>Text</i>	5
Pareigos	<i>Text</i>	30
Telefonas_faksas	<i>Text</i>	30
El paštas	<i>Hyperlink</i>	25

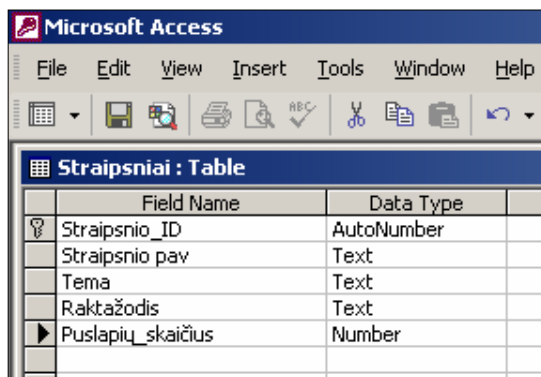


Field Name	Data Type
Recenzento_ID	AutoNumber
Vardas	Text
Pavardė	Text
Moksl_laipsn_pedag_moksl_v	Text
Organizacija	Text
Pareigos	Text
Telefonas_faksas	Text
El_paštas	Hyperlink

3.14 pav. Lentelės *Recenzantai* laukai ir laukų tipai

3.5 lentelė. Lentelės *Straipsniai* struktūra

Lauko pavadinimas	Lauko tipas	Lauko ilgis
Straipsnio_ID	<i>AutoNumber</i>	
Straipsnio_pavadinimas	<i>Text</i>	70
Tema	<i>Text</i>	30
Raktažodis	<i>Text</i>	35
Puslapių_skaičius	<i>Number</i>	<i>Long Integer</i>

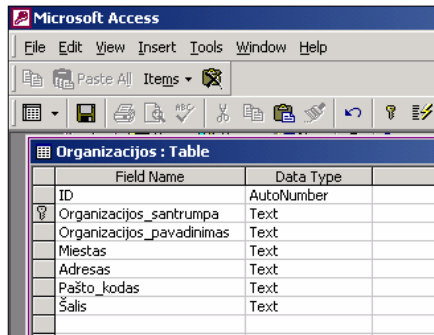


Field Name	Data Type
Straipsnio_ID	AutoNumber
Straipsnio_pav	Text
Tema	Text
Raktažodis	Text
Puslapių_skaičius	Number

3.15 pav. Lentelės *Straipsniai* laukai ir laukų tipai

3.6 lentelė. Lentelės *Organizacijos* struktūra

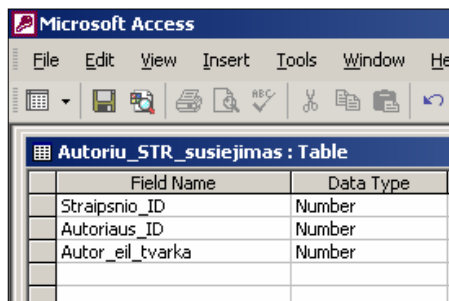
Lauko pavadinimas	Lauko tipas	Lauko ilgis
ID	<i>AutoNumber</i>	
Organizacijos santrumpa	<i>Text</i>	5
Organizacijos pavadinimas	<i>Text</i>	35
Miestas	<i>Text</i>	15
Adresas	<i>Text</i>	45
Pašto kodas	<i>Text</i>	10
Šalis	<i>Text</i>	20



3.16 pav. Lentelės *Organizacijos* laukai ir laukų tipai

3.7 lentelė. Siejamosios lentelės *Autorių_STR_susiejimas* struktūra

Lauko pavadinimas	Lauko tipas	Lauko ilgis
Straipsnio_ID	<i>Number</i>	<i>Long Integer</i>
Autoriaus_ID	<i>Number</i>	<i>Long Integer</i>
Autorių_eil_tvarka	<i>Number</i>	<i>Byte</i>



3.17. pav. Autorių ir straipsnių siejamosios lentelės laukai ir laukų tipai

Laukų pavadinimus gali sudaryti ne tik raidės, bet ir skaitmenys, tarpai bei specialieji simboliai, išskyrus kablelį, šauktuką, tašką ir laužtinius skliaustus. Pavadinimai negali prasidėti tarpu, didžiausias leidžiamas pavadinimo ilgis – 64 simboliai.

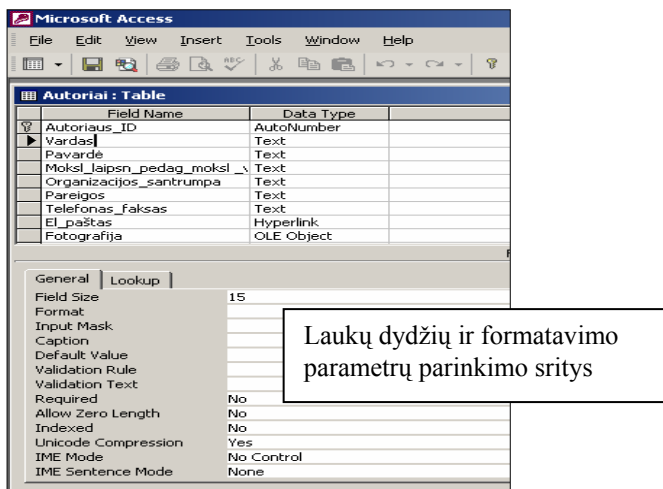
Svarbu tinkamai parinkti ir teisingai įvesti lauko pavadinimą, nustatyti įvedamų duomenų tipą ir lauko ilgį, nuo to priklausys vėlesnis darbas ir įvedamų duomenų kokybė.

Jeigu lentelėje pateikti laukų pavadinimai, duomenų tipai ir laukų ilgis netenkina, juos galima keisti. Galima keisti ir lentelių pavadinimus (*Rename*).

Kaip nustatomi laukų dydžiai ir jų formatai?

Sudarius lentelės projektą reikia nurodyti kiekvienam laukui priskiriamus minimizuojančius parametrus:

- ✚ Laukų dydžius (*Field Size*), nustatant maksimalų skiriamą atitinkamo tipo duomenų lauko skilčių skaičių;
- ✚ Įvedamų duomenų formatus (*Format*);
- ✚ Įvedimo formas (*Input Mask*);
- ✚ Įrašomas pradinės reikšmės (*Default Value*);
- ✚ Lauko reikšmių tikrinimo parametrus (*Validation Rule*);
- ✚ Indeksavimo savybes (*Indexed*) ir kt.

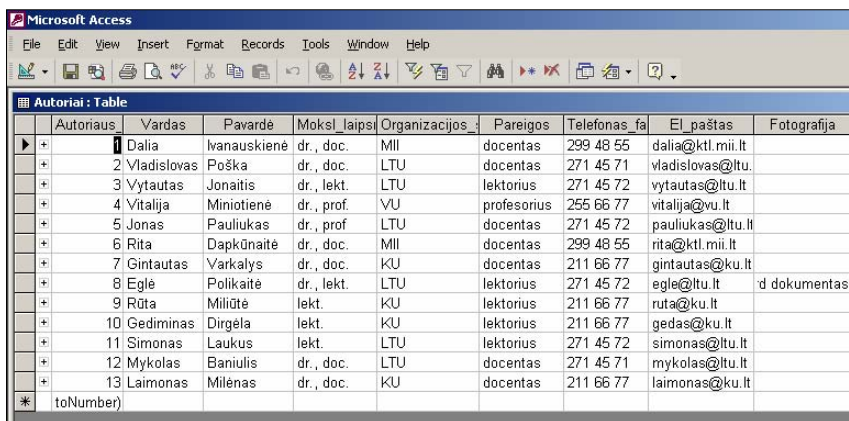


3.18 pav. Aktyvaus lauko formatavimo parametrų parinkimas

Kaip duomenis įvesti į lentelę?

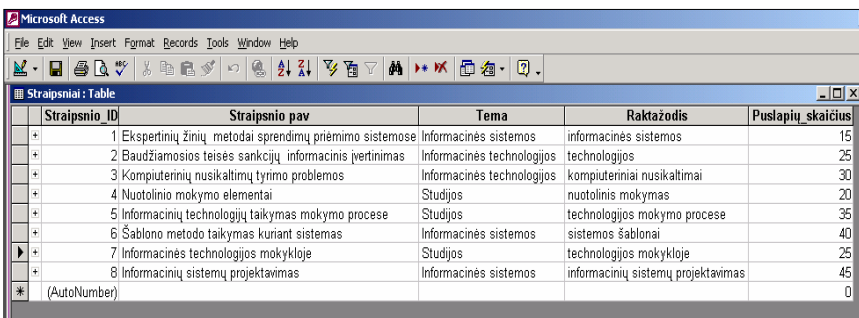
Duomenis į lentelę galima įvesti pasirenkant objektą *Table* ir dukart spragtelint kairiuoju pelės klavišu ties lentelės pavadinimu. Darbo lauke atsiras lentelė, kurią bus galima pildyti. Užpildžius arba norint nutraukti pildyti reikia uždaryti lentelės (*Table*) langą.

Kai kurių laukų tipų duomenys (pvz., *OLE objects*, *MEMO*) gali būti įvedami tik sukūrus duomenų įvedimo formą. Pavyzdžiui, jeigu norime įvesti *fotografiją*, t. y. *OLE* tipo lauką, sukuriame atitinkamą duomenų įvedimo formą (*Forms*). Jeigu patogiau dirbti su duomenų įvedimo forma, tai duomenis į lentelę galima įvesti naudojant formas.



	Autoriaus	Vardas	Pavardė	Moksl.laipsnis	Organizacijos	Pareigos	Telefonas	f.a	El. paštas	Fotografija
+	1	Dalia	Ivanauskienė	dr., doc.	MII	docentas	299 48 55		dalia@ktl.mii.lt	
+	2	Vladislovas	Poška	dr., doc.	LTU	docentas	271 45 71		vladislovas@ltu.	
+	3	Vytautas	Jonaitis	dr., lekt.	LTU	lektorius	271 45 72		vytautas@ltu.lt	
+	4	Vitalija	Miniotienė	dr., prof.	VU	profesorius	255 66 77		vitalija@vu.lt	
+	5	Jonas	Pauliukas	dr., prof.	LTU	docentas	271 45 72		pauliukas@ltu.lt	
+	6	Rita	Dapkūnaitė	dr., doc.	MII	docentas	299 48 55		rita@ktl.mii.lt	
+	7	Gintautas	Varkalys	dr., doc.	KU	docentas	211 66 77		gintautas@ku.lt	
+	8	Eglė	Polikaitė	dr., lekt.	LTU	lektorius	271 45 72		egle@ltu.lt	d dokumentas
+	9	Rūta	Miliūtė	lekt.	KU	lektorius	211 66 77		ruta@ku.lt	
+	10	Gediminas	Dirgėla	lekt.	KU	lektorius	211 66 77		gedas@ku.lt	
+	11	Simonas	Laukus	lekt.	LTU	lektorius	271 45 72		simonas@ltu.lt	
+	12	Mykolas	Baniulis	dr., doc.	LTU	docentas	271 45 71		mykolas@ltu.lt	
+	13	Laimonas	Milėnas	dr., doc.	KU	docentas	211 66 77		laimonas@ku.lt	

3.19 pav. Lentelės *Autoriai* duomenys



	Straipsnio ID	Straipsnio pav	Tema	Raktazodis	Puslapių skaičius
+	1	Ekspertinių žinių metodai sprendimų priėmimo sistemose	Informacinės sistemos	informacinės sistemos	15
+	2	Baudžiamosios teisės sankcijų informacinis įvertinimas	Informacinės technologijos	technologijos	25
+	3	Kompiuterinių nusikaltimų tyrimo problemos	Informacinės technologijos	kompiuteriniai nusikaltimai	30
+	4	Nuotolinio mokymo elementai	Studijos	nuotolinis mokymas	20
+	5	Informacinių technologijų taikymas mokymo procese	Studijos	technologijos mokymo procese	36
+	6	Šablono metodo taikymas kuriant sistemas	Informacinės sistemos	sistemos šablonai	40
+	7	Informacinės technologijos mokykloje	Studijos	technologijos mokykloje	25
+	8	Informacinių sistemų projektavimas	Informacinės sistemos	informacinių sistemų projektavimas	45

3.20 pav. Lentelės *Straipsniai* duomenys

Kontroliniai klausimai

1. Kokias funkcijas atlieka duomenų bazių valdymo sistema?
2. Išvardykite jums žinomas duomenų bazių valdymo sistemas?
3. Kokios svarbiausios DBVS *Microsoft Access* dalys?
4. Išvardykite pagrindinius darbo su duomenų baze etapus.
5. Kaip sudaromas lentelės projektas?
6. Kaip redaguojamas lentelės projektas?
7. Kokie galimi DBVS *MS Access* duomenų tipai?
8. Kokio tipo duomenys nesaugomi lentelėse?
9. Kaip duomenis įvesti į lentelę?

3.7.2. Lentelių ryšių nustatymas duomenų bazėje

1. Darbo tikslas

Išmokti sudaryti lentelių ryšių projektą, išsiaiškinti ryšio savybes bei atmainas, mokėti redaguoti ryšius.

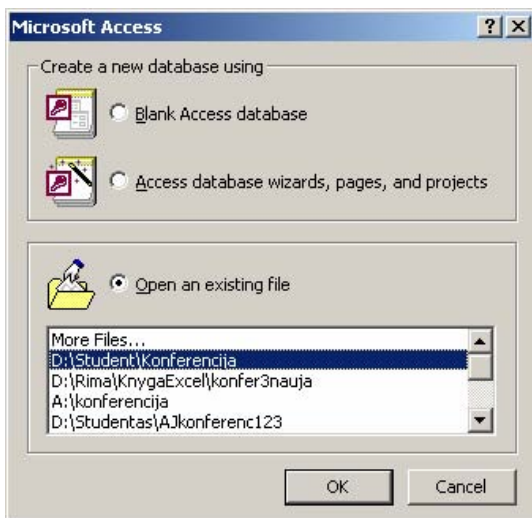
2. Praktinio darbo užduotys

1. Iškviškite jau pradėtą kurti konferencijos duomenų bazę.
2. Išsiaiškinkite, koku tikslu nustatomi lenteles siejantys ryšiai.
3. Pasirinkite lenteles, tarp kurių nustatysite ryšius.
4. Nustatykite ryšį tarp tą pačią prasmę turinčių laukų, esančių skirtingose lentelėse.
5. Išsiaiškinkite ryšio savybes bei mokėkite jį redaguoti dirbdami ryšio projekto lange.
6. Išsiaiškinkite, kokios gali būti ryšio atmainos.
7. Išsaugokite sudarytąjį lentelių ryšių projektą, išmokite jį iškviešti ir redaguoti.

3. Metodiniai nurodymai

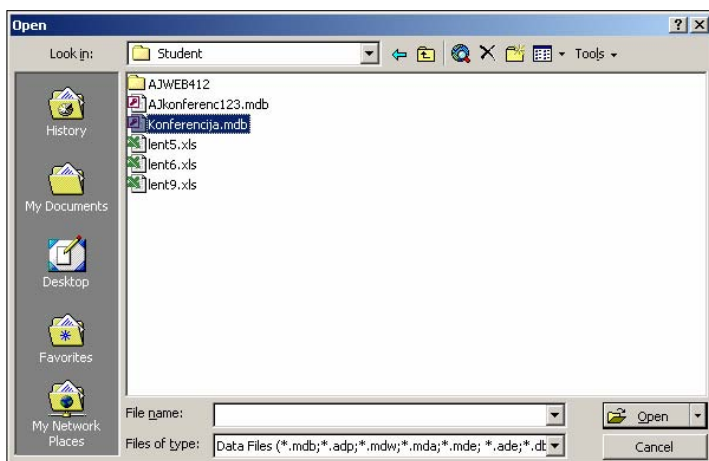
Kaip iškviešti dirbti egzistuojančią duomenų bazę?

Iškviškite programą *Microsoft Access*, dialogo langelyje *Microsoft Access* suaktyvinkite *Open an existig file*, pažymėkite duomenų bazės pavadinimą ir ją atverkite.



3.21 pav. Diske esančios duomenų bazės atvėrimo langas

Jeigu šiame dialogo langelyje jūsų konferencijos duomenų bazės pavadinimo nėra, suaktyvinkite *More Files*, dialogo langelyje *Open* suraskite duomenų bazę ir ją atverkite.



3.22 pav. Duomenų bazės pasirinkimas, norint ją atverti

Lentelės ryšių samprata

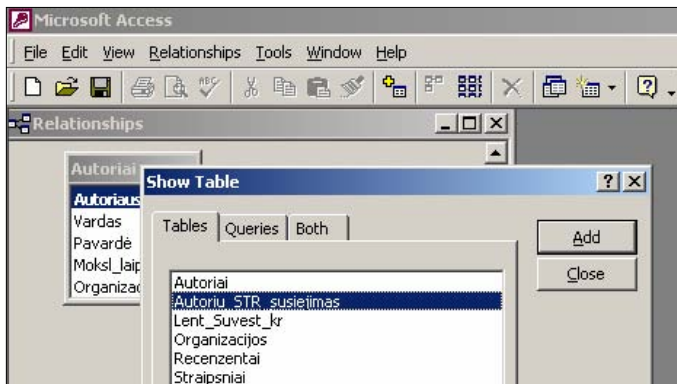
Duomenis saugoti lentelėse yra patogiu, su jais lengva ir paprasta dirbti, nes informacija išskaidoma į kelias lenteles. Norint pateikti informaciją iš kelių lentelių, reikia nustatyti lenteles siejančius ryšius, t. y. apibrėžti laukų, esančių skirtingose lentelėse ir turinčių tą pačią prasmę, ryšius (plačiau galima susipažinti 3.3. skyriuje).

Parengus duomenų bazės lentelių projektą labai svarbus darbo etapas yra nustatyti skirtingose lentelėse esančių ir tas pačias reikšmes turinčių laukų ryšius. Tai palengvina duomenų paiešką, padeda remiantis kelių lentelių duomenimis sudaryti užklausas ir ataskaitas.

Lentelių ryšių nustatymas

Norint nustatyti lentelių ryšius reikia:

- ✚ Vykdyti komandą *Tools* → *Relationships*, kuri iškviečia komandos projekto langą *Relationships* ryšiams tarp lentelių sudaryti;
- ✚ Vykdyti komandą *Relationships* → *Show Table*, dialogo lange *Show Table* leidžiančia pasirinkti lenteles, tarp kurių sudaromi ryšiai.

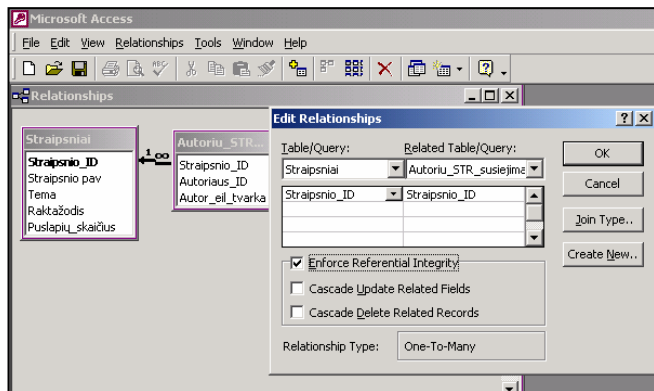


3.23 pav. Lentelių ryšių nustatymo dialogo langas

Lentelių ryšiai sudaromi ryšių projekto lange *Relationships*. Šiame projekto lange reikia patogiai išdėstyti lenteles, kad sudarytas lentelių ryšio projektas būtų aiškus.

Laukus siejančių ryšių nustatymas

Sudarant ryšį tarp skirtingose lentelėse esančių, tačiau tą pačią prasmę turinčių laukų, pažymimas vienas laukas, spaudžiant kairįjį pelės klavišą pažymimas kitas laukas. Atleidus pelės klavišą pateikiamas ryšio redagavimo langas *Edit Relationships*. Parenkamasis ryšio tipas: *vienas su vienu* (1:1), *vienas su daugeliu* (1:N), arba *daugelis su vienu* (N:1). Ryšiai reliacinėse duomenų bazėse, nesinaudojant pirminiais raktais, formuojami gana retai.



3.24 pav. Lentelių ryšių redagavimo langas

Ryšių redagavimas

Redaguoti ryšį galima ties jau suformuoto ryšio linija spaudžiant dešinįjį pelės klavišą ir vykdant kontekstinio meniu komandą *Edit Relationships*.

Ryšių redagavimo lange *Edit Relationships* galima pažymėti šias ryšių savybes:

- + Ryšio integralumas (*Enforce Referential Integrity*). Ryšio integralumas rodo, kad išorinio rakto lauke negali būti tokių lauko reikšmių, kurių nėra pagrindinės lentelės valdančiame lauke. Pateiktame pavyzdyje ryšio redagavimo lange *Edit Relationships* valdantysis laukas yra *Straipsnio_ID* iš lentelės *Straipsniai*, išorinio rakto laukas yra *Straipsnio_ID* iš lentelės *Autorių_STR_susiejimas*.
- + Kaskadinis susietų laukų atnaujinimas (*Cascade Update Related Fields*). Pavyzdžiui, jeigu lentelėje *Straipsniai* pa-

keičiamas straipsnio identifikatorius, jis atnaujinamas ir lentelėje *Autorių_STR_susiejimas*.

- ✚ Kaskadinis susietų įrašų naikinimas (*Cascade Delete Related Records*). Pavyzdžiui, pašalinus *Straipsnio_ID* iš lentelės *Straipsniai*, straipsniai bus šalinami ir iš lentelės *Autorių_STR_susiejimas*.

Ryšio atmainos parinkimas

Ryšio redagavimo lange *Edit Relationships* mygtuku *Join Type* galima parinkti ryšio atmainą (kardinalumo apribojimus). Nuspaudus mygtuką *Join Type* pateikiamas dialogo langas *Join Properties*. Jame matyti trys galimos ryšio atmainos.



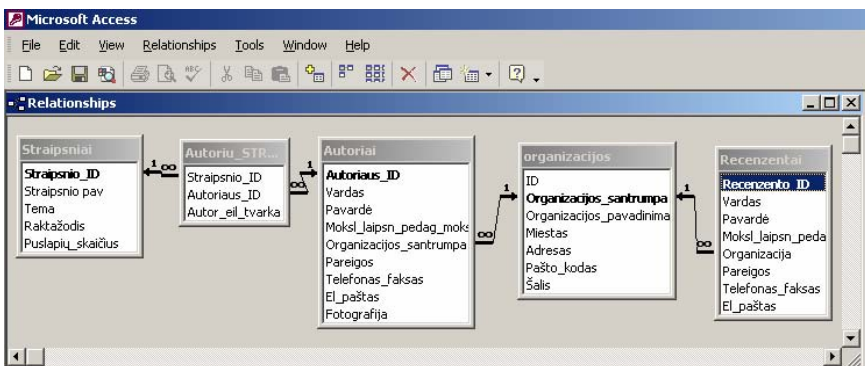
3.25 pav. Ryšio savybių parinkimo langas

Parinktoji lentelių ryšio atmaina nurodo lentelių pavaldumo kryptį ir susijusių įrašų atrinkimo būdą užklausoje:

1. Ryšys apima tik tuos įrašus, kuriuose laukų reikšmės sutampa;
2. Ryšys apima visus pirmosios lentelės ir tuos antrosios lentelės įrašus, kurių ryšio laukų reikšmės yra pirmoje lentelėje;
3. Ryšys apima visus antrosios lentelės ir tuos pirmosios lentelės įrašus, kurių ryšio lauko reikšmės yra antroje lentelėje.

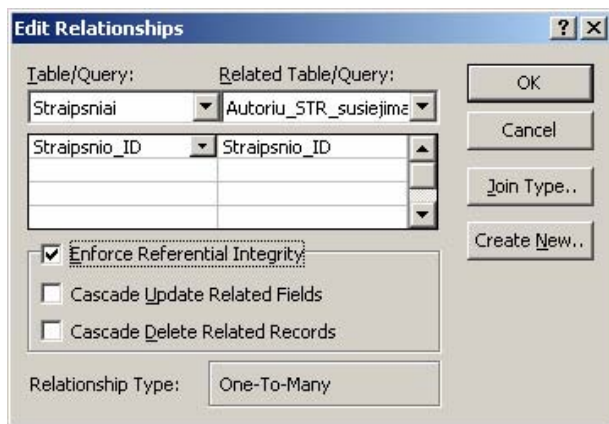
Lentelių ryšių projekto langas

Ryšių rodyklės matyti, kai yra įjungtas ryšio integralumo požymis (*Enforce Referential Integrity*).



3.26 pav. Lentelių ryšių projekto langas

Redagavimo lange *Edit Relationships* parinktos ryšio savybės gali būti keičiamos, pakeitimai patvirtinami mygtuku *Ok*.



3.27 pav. Ryšio savybės keitimo langas

Redaguojant ryšių tipai keičiami išskviečiant papildomų ryšio savybių langą. Tai atliekama spaudžiant *Join Type* mygtuką.



3.28 pav. Ryšio savybių parinkimo langas

Kaip išsaugoti sudarytą lentelių ryšio projektą?

Duomenų bazės ryšių projektas įrašomas į kuriamos duomenų bazės failą. Esant ryšių projekto lange reikia paspausti priemonių juostoje esantį mygtuką *Save* arba vykdyti ekvivalentiškas išsaugojimo komandas. Tačiau ir uždarant *MS Access* langą visi aprašymai išlieka duomenų bazės faile.

Kontroliniai klausimai

1. Kokia komanda pradedamas kurti lentelių ryšio projektas?
2. Tarp kokių lentelių laukų paprastai sudaromi ryšiai?
3. Kokius ryšių tipus žinote ir kokia jų prasmė?
4. Kam reikalingi skirtingų lentelių ryšiai?
5. Kaip pakeisti jau sukurtą ryšio projektą?
6. Kaip ryšiai redaguojami?
7. Kas yra ryšio integralumas, kokios yra kitos ryšio savybės?

3.7.3. Duomenų įvedimo formos ir jų projektavimas

1. Darbo tikslas

Išmokti sukurti patogią duomenų įvedimo formą ir įvesti bei redaguoti duomenis naudojantis bazėse esančiomis duomenų įvedimo formomis.

2. Praktinio darbo užduotys

1. Iškvieskite jau pradėtą kurti konferencijos duomenų bazę.
2. Sukurkite autorių duomenų įvedimo formą.
3. Pakeiskite autorių duomenų įvedimo formos projektą.
4. Į autorių duomenų formas įkelkite autorių nuotraukas.
5. Išmokite redaguoti duomenis naudodami duomenų įvedimo formas.
6. Išmokite rinktis, pridėti ir užpildyti įrašą ir jį ištrinti.
7. Sukurkite straipsnių duomenų įvedimo formą.

3. Metodiniai nurodymai

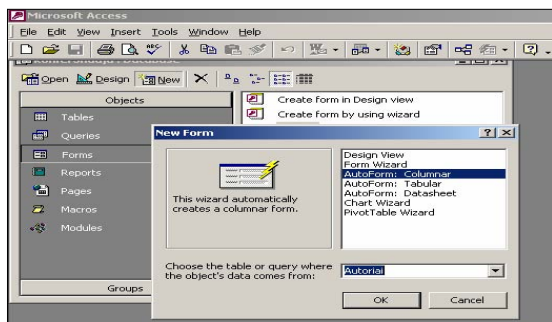
Kam reikalingos duomenų įvedimo formos?

Didelėms lentelėms tvarkyti patogiu naudoti duomenų įvedimo formas, kurios ekrane pateikia vieną įrašą, ir suteikia galimybę jį redaguoti bei pereiti prie kito įrašo. Formą galima kurti savarankiškai bei naudojantis vedliu (*Wizard*).

Formos (*Forms*) leidžia pasirinkti patogią formą duomenims įvesti, juos redaguoti ir peržiūrėti, jos naudojamos įterpti *OLE* objektams, pavyzdžiui, nuotraukoms, duomenis iliustruojantiems paveikslėliams, *MEMO* laukams ir panašiai.

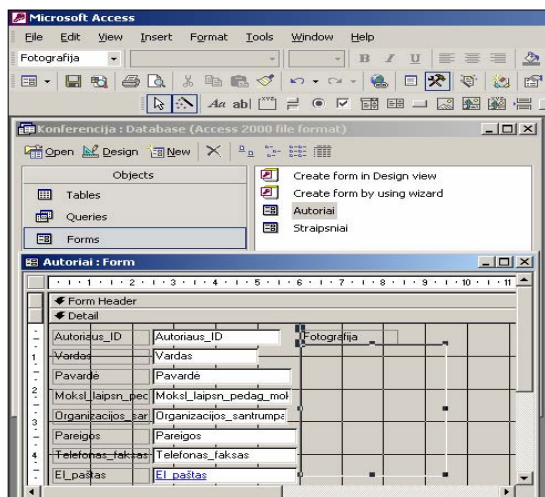
Kaip kurti duomenų įvedimo formos projektą?

Suaktyvinkite objektą *Forms*, paspauskite mygtuką *New*, dialogo langelyje *New Form* pasirinkite formos tipą, pavyzdžiui, *Auto-Form: Columnar*, dalyje *Chose the table or query where...* iš sąrašo pasirinkite, iš kokių duomenų formą sudarote, pavyzdžiui, lentelę *Autoriai*.



3.29 pav. Duomenų įvedimo formos kūrimo langas

Jeigu duomenų įvedimo formos projektas nėra patogus dirbti su duomenimis, jį galite pakeisti. Esant pažymėtai formai, pavyzdžiui, *Autoriai*, spauskite projektavimo mygtuką *Desing*. Galite pakeisti nuotraukos dydį bei praplėsti matomas laukų ribas, perkelti laukus į patogią dirbti vietą.

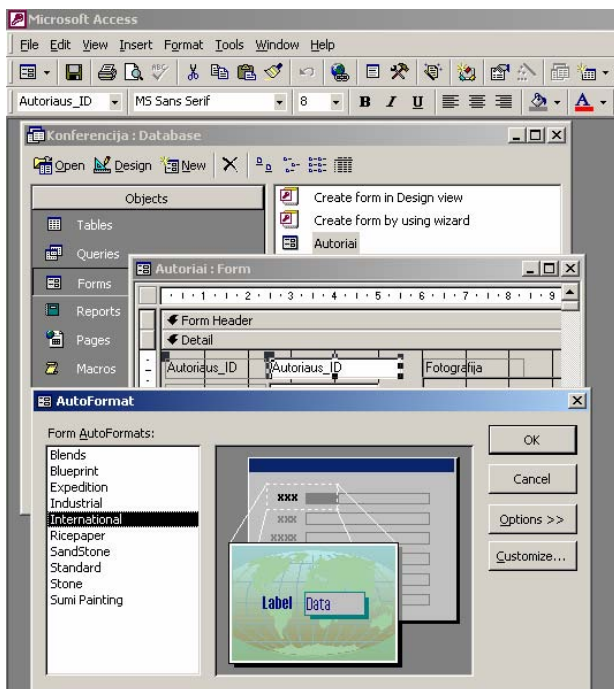


3.30 pav. Duomenų įvedimo formos projektavimo langas

Formos laukus galima apipavidalinti automatiškai: kai matyti formos projekto langas, reikia vykdyti komandą *Format* → *AutoFormat* ir pasirinkti pažymėtų laukų formatavimo tipą.

Formą galima papildyti paveikslais ir piešiniais. Jeigu esant aktyviam projekto langui įvykdoma komanda *Insert*→*Picture* ir failo užklauskos lange nurodomas piešinio failo pavadinimas, į formą įkeliamas piešinys. Formoje matomą piešinį galima perkelti į norimą vietą bei pakeisti jo matmenis.

Naudojant pagalbinę atmintį *Clipboard* piešinius galima tiesiogiai perkelti iš kitų programų.

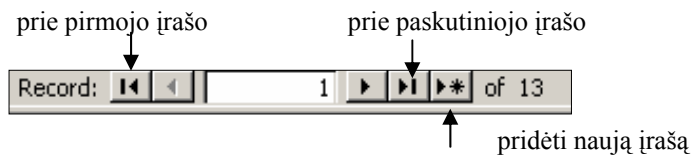


3.31 pav. Formos laukų apipavidalinimas vykdant komandą *Format*→*AutoFormat*

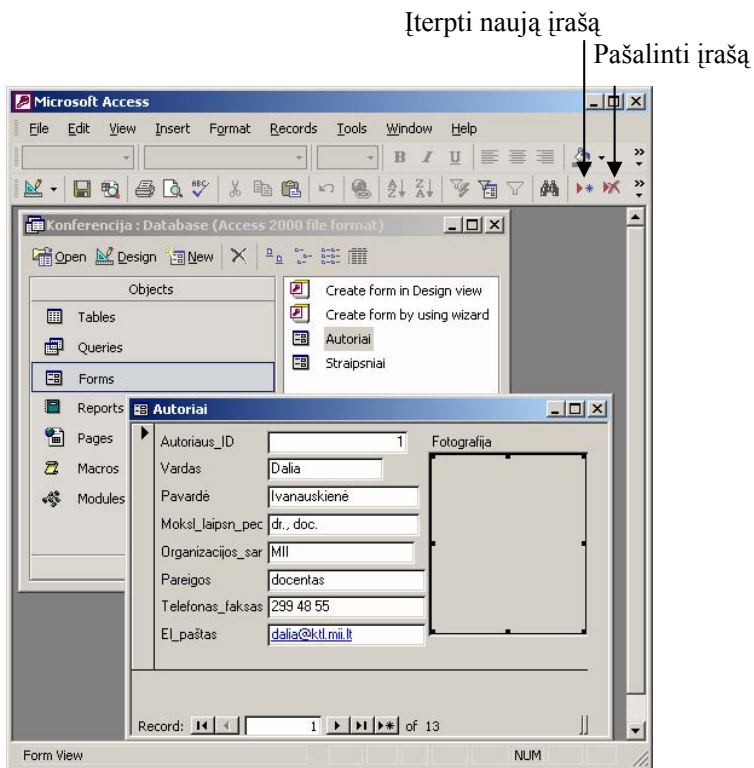
Kaip naudoti formą, skirtą duomenims įvesti bei jiems redaguoti?

Du kartus spaudžiant kairiąją pelės klavišą ties formos pavadinimu atveriamas forma, kurioje yra jai priskirtos lentelės duomenys, kuriuos galima tvarkyti standartinėmis tekstų redagavimo priemonėmis. Judėti iš vieno lauko į kitą žemyn galima spaudžiant klavišus <Enter> arba <Tab>, o į viršų – klavišus <Shift>+<Tab>.

Judėti nuo įrašo prie įrašo galima šiais mygtukais:



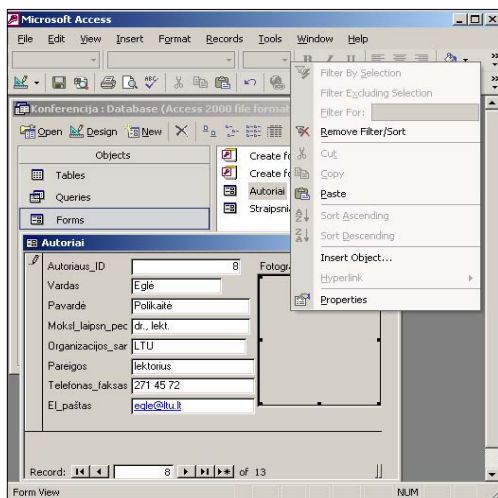
3.32 pav. Lentelės įrašų perrinkimo valdymo mygtukai



3.33 pav. Formos *Autoriai* pirmojo įrašo duomenų langas

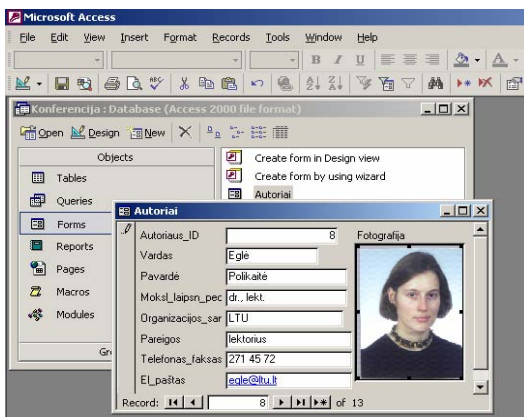
Įterpti nuotrauką į *OLE* tipo lauką galima komanda *Paste* iškviečiant kontekstinį fotografijos lauko meniu, pasitelkiant pagalbinę iškarpų atmintį *Clipboard*.

Į *OLE* tipo lauką galima įterpti duomenis ir iš failo. Tuomet vykdoma komanda *Insert Object* ir standartiniu būdu parenkamas failas.

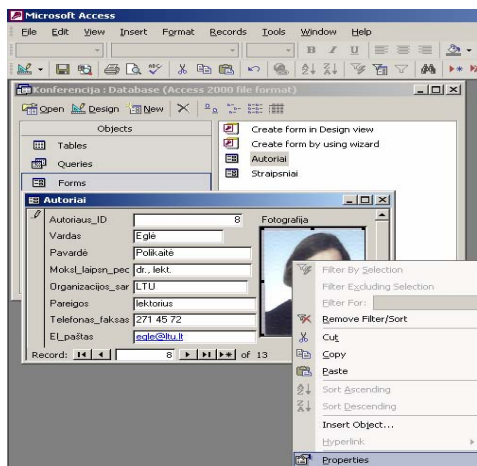


3.34 pav. Fotografijos įterpimo kontekstinis meniu

Naudojantis formomis galima laisvai judėti nuo įrašo prie įrašo, įterpti nuotraukas. Svarbu, kad nuotraukos dydis atitiktų formoje laukui skirtą dydį. Esant dydžių skirtumui reikia keisti formos projektą arba įterpiamos nuotraukos dydį.

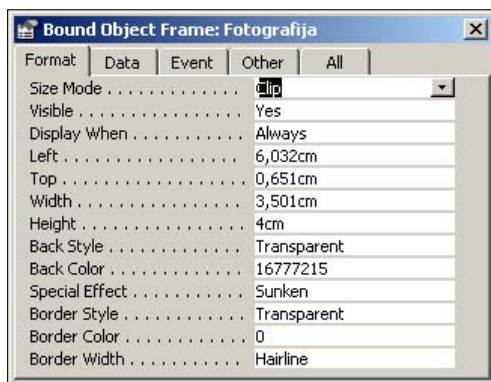


3.35 pav. Formos papildymas fotografija



3.36 pav. Fotografijos parametru keitimas vykdamas komanda *Properties*

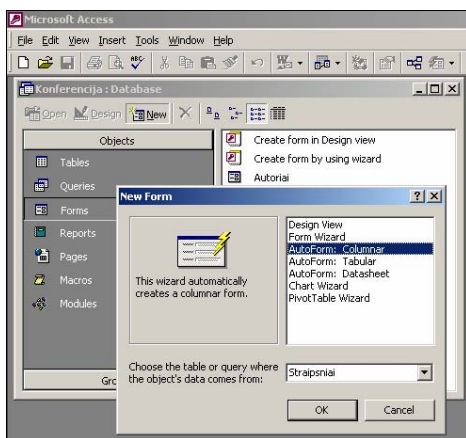
Įvykdžius kontekstinio meniu komanda *Properties* OLE tipo objektui skirtame lauke gaunamas dialogo langelis *Bound Object Frame: Fotografija*, skyrelyje *Format* galima parinkti tinkamus apipavidalinimo parametrus, pavyzdžiui, aukštį, plotį, specialius efektus, rėmeliu stilių, išdėstymą ir panašiai.



3.37 pav. OLE tipo objekto apipavidalinimo parametru parinkimo langas

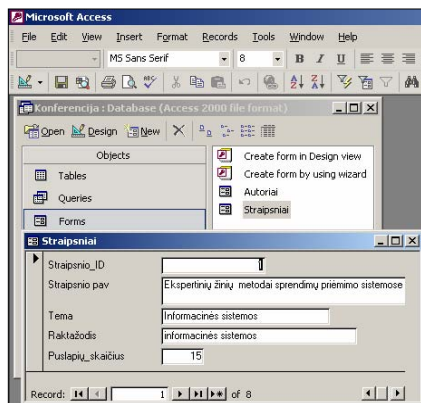
Kaip sukurti straipsnių registracijos duomenų įvedimo formą?

Straipsnių duomenų įvedimo forma kuriama analogiškai kaip ir autorių duomenų įvedimo forma. Esant aktyviam *Forms* objektui mygtuku *New* vykdoma komanda, skirta naujai formai įvesti. Įvykdžius komandą dialogo lange *New Form* parenkamas formos tipas bei šaltinis, iš kurio bus imami pradiniai duomenys, t. y. jau egzistuojanti lentelė *Straipsniai*.



3.38 pav. Duomenų įvedimo formos *Straipsniai* kūrimas

Forma su jai priskirtos lentelės pirmojo įrašo duomenimis (3.39 pav.) atveriamas suaktyvinant objektą *Forms* ir pasirenkant sukurto formos pavadinimą, pavyzdžiui, *Straipsniai*.



3.39 pav. Formos *Straipsniai* pirmasis įrašas

Kontroliniai klausimai

1. Kokius žinote lentelės duomenų įvedimo ir redagavimo formų kūrimo būdus?
2. Kaip pakeisti jau sukurtos formos projektą?
3. Kokius veiksmus reikia atlikti į formą įkeliant grafinį vaizdą?
4. Kaip naudojant formą galima įterpti naują lentelės įrašą arba pašalinti jame esantį?
5. Kokio tipo laukai yra matomi formose, tačiau nematomi lentelėse?

3.7.4. Duomenų paieška, rūšiavimas ir filtravimas duomenų bazėje

1. Darbo tikslas

Išmokti atlikti paiešką *Microsoft Access* duomenų bazėje, taikant įvairius paieškos būdus bei sąlygas.

Išsiaiškinti, kaip panaudoti skirtingus duomenų atrankos (filtravimo) būdus: filtravimą pagal požymius, filtravimą taikant sąlygas ir filtro pertvarkymą į užklausą.

2. Darbo užduotys

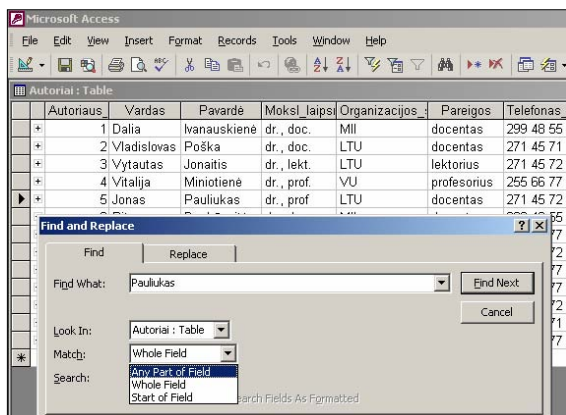
1. Iškvieskite programą *MS Access*. Atverkite sukurtą duomenų bazę.
2. Atlikite duomenų paiešką konkrečioje lentelėje (pvz., *Autoriai*) pagal konkretų lauko pavadinimą (pvz., *Organizacija*: „VU“ ir pan.).
3. Susipažinkite su įvairiais paieškos būdais.
4. Atlikite duomenų filtravimą konkrečioje lentelėje (pvz., iš lentelės *Autoriai* išrinkite visus vienos organizacijos autorius).
5. Atlikite duomenų filtravimą pagal filtravimo sąlygas, nustatydami santykio operatorius.
6. Sudarykite filtrą naudodami duomenų formas.
7. Suprojektuokite keletą naudojimosi išplėstiniu filtru būdų ir išsaugokite juos kaip užklausas.

3. Metodiniai nurodymai

Kaip atlikti paiešką, norint surasti duomenis?

Norint atlikti paiešką atveriamą lentelė ir vykdoma komanda *Edit*→*Find*. Komandos dialogo lange *Find and Replace* nurodoma, ką reikia rasti (*Find what*), pavyzdžiui, autorių, kurio pavardė – Pauliukas. Išskleidžiamame sąraše išrenkamas tinkamas lyginimo būdas (*Match*):

- ✚ Norint, kad sutaptų bet kuri lauko reikšmės dalis, pasirenkama *Any Part of Field*;
- ✚ Norint, kad sutaptų visas laukas, – *Whole Field*;
- ✚ Norint, kad sutaptų lyginamojo lauko pradžia, – *Start of Field*.

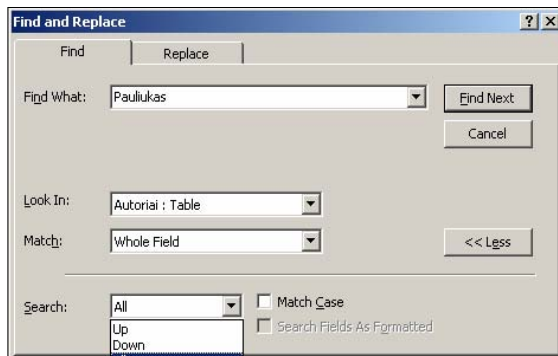


3.40 pav. Komandos *Edit*→*Find* paieškos sąlygų langas

Paspaudus paieškos mygtuką (*Find Next*) programa lentelėje suranda pirmą atitinkantį sąlygas įrašą. Dar kartą paspaudus mygtuką *Find Next* programa suranda kitą sąlygas tenkinantį įrašą arba pateikia pranešimą, kad tokio įrašo nėra.

Spaudžiant dialogo lango *Find and Replace* mygtuką *More*>> paieškos sąraše (*Search*) galima pasirinkti paieškos kryptį: aukštyn (*Up*), žemyn (*Down*), visoje lentelėje (*All*).

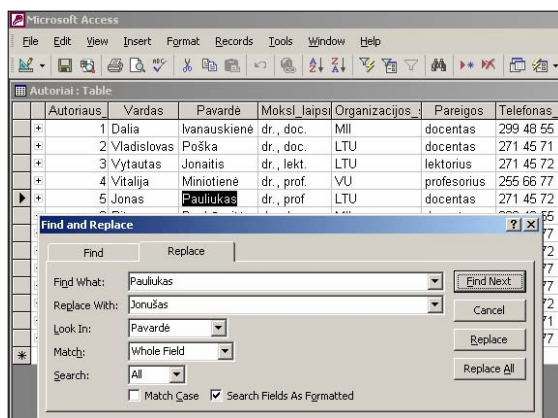
Jeigu reikia skirti didžiąsias ir mažąsias raides, įjungiamas jungiklis *Match Case*.



3.41 pav. Komandos *Edit*→*Find* paieškos krypties pasirinkimo būdas

Kaip atlikti duomenų paiešką ir pakeisti juos kitais duomenimis?

Norint lentelėje surasti duomenis ir juos pakeisti kitais, teksto įvedimo žymiklį reikia nustatyti reikiamame duomenų stulpelyje ir vykdyti komandą *Edit*→*Replace*. Pateikto dialogo lango *Find and Replace* dalyje *Replace What* nurodykite, ką keisite, o lauko dalyje *Replace With* nurodykite naują lauko reikšmę.



3.42 pav. Duomenų paieškos ir keitimo langas

Įvairesnes paieškos sąlygas galima sudaryti naudojant šabloną (metasimbolius, kaukę):

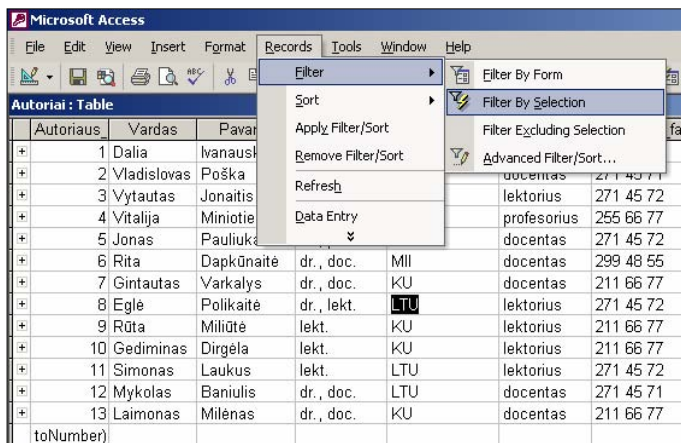
* – bet kokiai simbolių grupei surasti;

? – vienam simboliui surasti.

Pavyzdžiui, *is – ieškoma visų duomenų, turinčių galūnę -is, ?????? – ieškoma duomenų, susidedančių iš šešių simbolių.

Kaip atlikti lentelės duomenų filtravimą išrenkant juos pagal pažymėtas reikšmes?

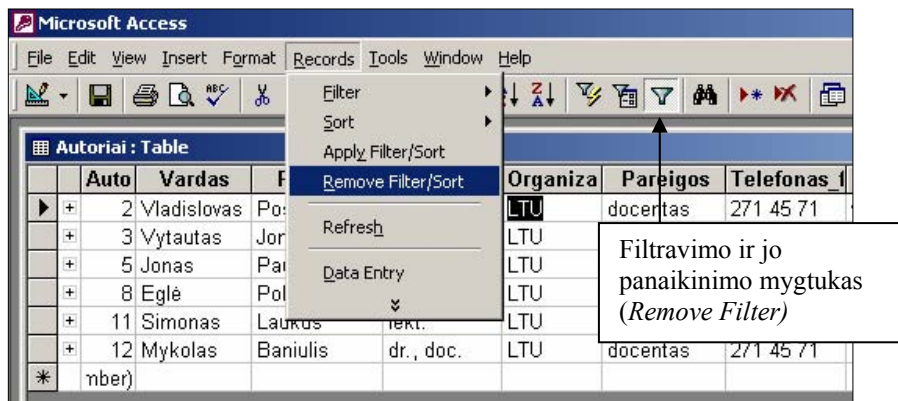
Norint iš lentelės išrinkti norimus duomenis, filtruojama pagal atitinkamą lauko reikšmę. Vienas iš būdų filtravimui atlikti yra pažymėti reikiamą duomenį lentelėje (pvz., LTU) ir vykdyti komandą *Record*→*Filter*→*Filter by Selection*.



3.43 pav. Duomenų filtravimas lentelėje pagal išrinktą

Įvykdžius komandą ekrane bus matyti įrašai, sutampantys su pažymėtają reikšmę. Atlikus lentelės filtravimą galima parinkti kitą atrankos kriterijų (naują filtrą pagal kito lauko reikšmę), taip bus galima atrinkti duomenis pagal kelių laukų reikšmes.

Norint, kad matytųsi visi pradiniai lentelės duomenys filtravimo atsisakoma, vykdant pagrindinio meniu komandą *Records*→*Remove*→*Filter/Sort* arba spaudžiant mygtuką *Remove Filter*.



3.44 pav. Filtravimo panaikinimo komandos

Kaip atlikti duomenų filtravimą lentelėje pagal sąlygas?

Filtravimą atliksime konkrečioje lentelėje, pavyzdžiui, *Autoriai*.

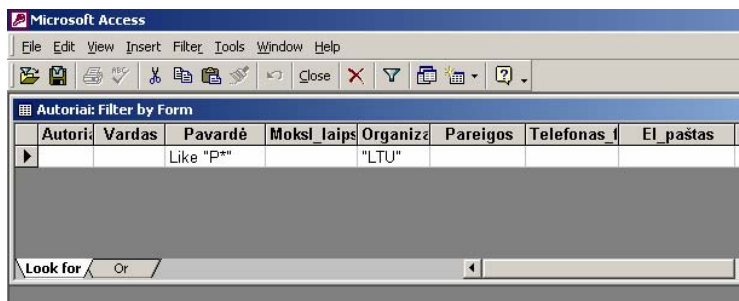
Norint atlikti filtravimą pagal filtro pateiktas reikšmes, vykdoma komanda *Record* → *Filter* → *Filter by Form*. Įvykdžius komandą atsiveria tuščia lentelė, kurioje reikia pasirinkti arba įrašyti filtro organizavimo priemones:

- ✚ Iš išskleidžiamų sąrašų parenkamos atrankos reikšmės;
- ✚ Naudojant tinkamus operatorius įrašomos sąlygos, pagal kurias bus filtruojami duomenys.

3.8 lentelė. Filtravimo sąlygų aprašų tipai

Filtro organizavimo priemonė		Filtravimo rezultatas
Aprašo tipas	Realizavimo priemonė	
Viso sutapimo	Ieškoma konkrečios lauko reikšmės	Rodomi įrašai, kurių reikšmės sutampa su filtro reikšme
Dalinio sutapimo	Operatorius <i>Like</i>	Rodomi įrašai, tenkinantys šablona, pavyzdžiui, <i>Like „P*“</i>
Atkarpos	Operatorius <i>Between</i>	Rodomi įrašai, kurių intervale yra lauko reikšmės, pavyzdžiui, <i>Between 3 and 7</i>
Santykio	Santykio operatoriai	Rodomi įrašai, tenkinantys santykį, pavyzdžiui, <i>>5</i>

Pavyzdžiui, turime išrinkti visus LTU autorius, kurių pavardės prasideda raide P, parašiusius straipsnių informacinių technologijų tematika.



3.45 pav. Komandos *Record*→*Filter*→*Filter by Form* dialogo langas ir filtravimo sąlygų užrašymo būdai

Gautojoje lentelėje atitinkamuose laukuose įrašykite arba parinkite šias filtravimo sąlygas:

- ✚ Pavardės lauke nurodykite sąlygą, kad pavardė prasidėtų raide, t. y. *Like* „P*“;
- ✚ Organizacijos santrumpoje lauke nurodykite reikiamą filtravimo reikšmę – „LTU“;
- ✚ Temos lauke nurodykite reikšmę „Informacinės technologijos“.

Įvykdę komandą gausite pagal nustatytas atrankos filtro sąlygas išrinktus duomenis.

Kaip atrinktus duomenis surikiuoti lentelėje?

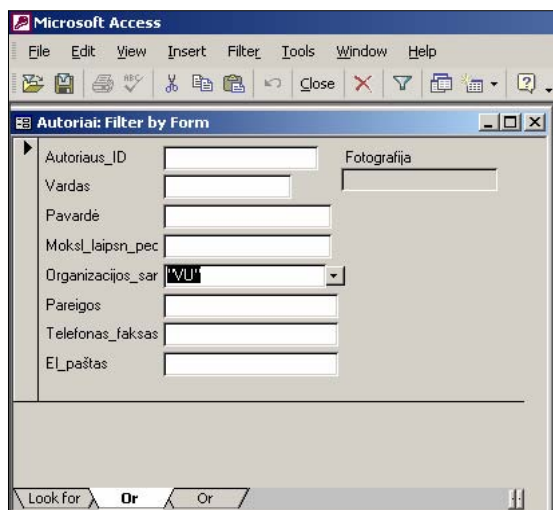
Norint filtruotus įrašus surikiuoti lentelėje reikia perkelti žymiklį į rikiavimo stulpelį ir vykdyti rūšiavimo komandas *Records*→*Sort*→*Sort Ascending* – didėjimo tvarka arba *Records*→*Sort*→*Sort Descending* – mažėjimo tvarka.

Norint surikiuoti duomenis pagal kelis požymius, reikia pažymėti šiuos požymius (laukų pavadinimus), laikant nuspaudus klavišą <Shift> ir kairiuoju pelės klavišu spragtelint ties laukų antraštėmis. Taip galima išrinkti tik gretimus laukus. Išrinktieji požymiai rikiuojami iš kairės į dešinę.

Kaip atlikti duomenų filtravimą naudojant duomenų formas?

Gali būti vykdomas ne tik duomenų bazės lentelių, bet ir formų duomenų filtravimas. Norint atlikti formų duomenų filtravimą reikia sukurti lentelės formą: pažymėti formos objektą ir dvigubu paspaudimu atverti duomenis, esančius formoje, pavyzdžiui, ankstesniame darbe sukurtą formą *Autoriai*.

Norint atlikti duomenų filtravimą išrenkant autorių iš LTU arba VU įrašų parenkama organizacijos santrumpa – LTU, formos apačioje spaudžiamas mygtukas *Or* ir parenkama organizacijos santrumpa VU. Vykdomas filtravimas ir gaunami duomenų įrašai, pateikti tam tikra forma.



3.46 pav. Duomenų filtravimo būdai formoje *Autoriai*

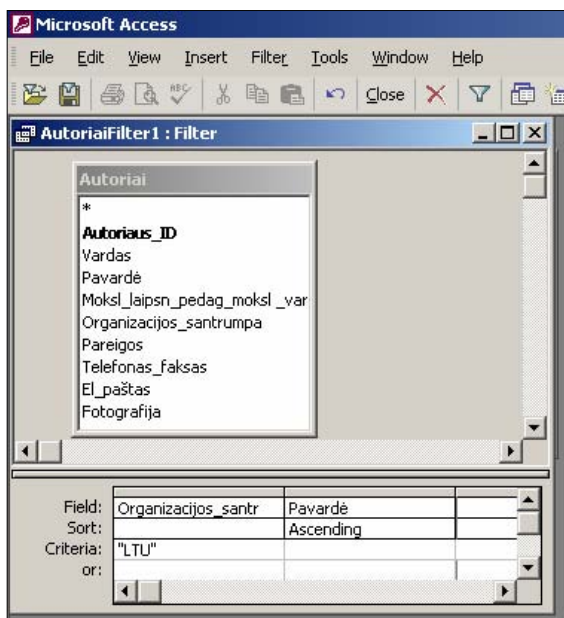
Kaip sukurti išplėstinį filtrą?

Išplėstinis filtravimas turi daugiau pranašumų:

- ✚ Galima pasirinkti keletą rikiavimo (rūšiavimo) būdų;
- ✚ Galima naudoti sudėtingas atrankos sąlygas (pagal daugelį rekvizitų);
- ✚ Filtrą galima išsaugoti kaip užklausą;
- ✚ Filtrą galima naudoti pakartotinai.

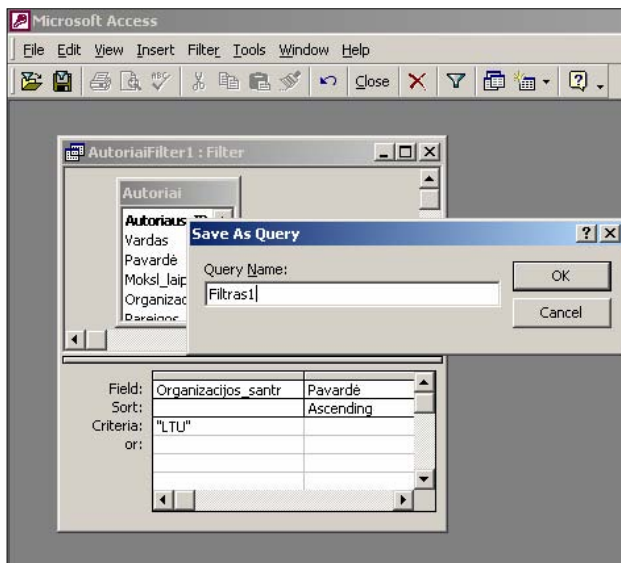
Norint sukurti išplėstinį autorių duomenų lentelės filtrą reikia suaktyvinti duomenų bazės objektą *Lentelės* ir atverti autorių duomenų lentelę. Esant atvertai autorių duomenų lentelei vykdoma komanda *Record*→*Filter*→*Advanced Filter/Sort*.

Sudarant išplėstinį filtrą atrankos ir rūšiavimo sąlygos aprašomos filtro projekto lange, nusakant kriterijus ir duomenų rūšiavimo būdą. *Criteria* „LTU“ – bus atrinkti visi įrašai, kuriuose yra organizacijos santrumpa *LTU*, duomenys atlikus filtravimą bus išdėstyti pagal pavardes abėcėlės tvarka (*Ascending*).



3.47 pav. Duomenų lentelės *Autoriai* išplėstinio filtro kūrimas

Filtrą galima išsaugoti kaip užklausą vykdant komandą *File*→*Save As Query* ir suteikti užklausiai pavadinimą, pavyzdžiui, *Filtras1*. Tuomet prireikus bus galima vykdyti užklausą pagal sudarytą filtrą (*Filtras1*) ir gauti duomenis, atrinktus pagal sudarytas filtravimo sąlygas.



3.48 pav. Filtras, kaip užklauskos išsaugojimo pavyzdys, vykdant komandą *File*→*Save As Query*, suteikiant pavadinimą *Filtras 1*

Dalyje *Criteria* užrašomos atrankos sąlygos. Jos gali būti nustatomos:

- ✚ Pagal konkrečią lauko reikšmę, pavyzdžiui, jeigu reikalingi konkrečios organizacijos autoriai, dalyje *Field* iš konkrečios lentelės pasirenkamas lauko pavadinimas, tarkime, *Autoriali*. Lauke *Organizacijos santrumpa* ir dalyje *Criteria* kabutėse įrašoma konkreti tekstinio lauko reikšmė, pavyzdžiui „LTU“;
- ✚ Pagal konkrečių lauko reikšmių disjunkciją ir konjunkciją (*OR*, *AND*);
- ✚ Santykio operatorių ir lyginamąją reikšmę: daugiau *>*, mažiau *<*, nelygu *<>* ir kt.

Pavyzdžiui, taip atrodo duomenys, išfiltruoti renkantis organizaciją LTU ir išdėstyti pagal pavardes abėcėlės tvarka.

	Autoria	Vardas	Pavardė	Moksl laipsn	Organizacijos_sant	Pa
+	12	Mykolas	Baniulis	dr., doc.	LTU	doc
+	3	Vytautas	Jonaitis	dr., lekt.	LTU	lekt
+	11	Simonas	Laukus	lekt.	LTU	lekt
+	5	Jonas	Pauliukas	dr., prof	LTU	doc
+	8	Eglė	Polikaite	dr., lekt.	LTU	lekt
+	2	Vladislavas	Poška	dr., doc.	LTU	doc
*	umber)					

3.49 pav. Duomenys, gauti atlikus filtravimą lentelėje *Autoriai*, išdėstyti abėcėlės tvarka

Galima filtruoti ir pagal logiką „nelygu“. Jeigu dalyje *Criteria* įrašytume $\langle \rangle$ „Baniulis“ ($\langle \rangle$ reiškia nelygu), būtų rodomi įrašai, kuriuose nėra pavardės Baniulis.

Pavyzdžiui, jeigu norime išrinkti ne LTU ir ne VU institucijų asmenis, lauko *Organizacijos_santr* dalyje *Criteria* turėtų būti nurodoma: $\langle \rangle$ „LTU“ AND $\langle \rangle$ „VU“. Toks filtras nerodo LTU ir VU autorių, todėl lieka tik kitų institucijų autoriai.

Kontroliniai klausimai

1. Kaip atliekamas lentelės duomenų filtravimas pagal išranką?
2. Kaip šalinamas lentelės duomenų filtras?
3. Ką reiškia filtravimo sąlyga: *Like „D*“*?
4. Kuo skiriasi formos pavidalo filtras nuo lentelės filtro?
5. Kaip sudaromas išplėstinis filtras?
6. Kokia komanda leidžia išsaugoti filtrą kaip užklausą?
7. Kokias sąlygas galima naudoti sudarant išplėstinį filtrą?
8. Kam yra skirtos išplėstinio filtro projekto sritys: *Field*, *Sort* ir *Criteria*?

3.7.5. Dialoginių užklausų atlikimo būdai

1. Darbo tikslas

Išsiaiškinti, kokie duomenų apdorojimo būdai atliekami vykdant užklausas, kokie yra užklausų tipai. Išmokti duomenų bazėje kurti dialogines užklausas.

2. Darbo užduotys

1. Iškvieskite programą *MS Access*. Atverkite sukurtą duomenų bazę.
2. Išsiaiškinkite, kokie yra užklausų tipai ir kokiais atvejais jie naudojami.
3. Atlikite dialoginę užklausą, pavyzdžiui, formuodami ją iš lentelių *Autoriai* ir *Organizacijos*. Užklausos tikslas – išsiaiškinti tikslų autoriaus adresą.
4. Susipažinkite su užklausos projekto langu bei projekto sudarymo būdais.
5. Susipažinkite su reiškinų konstruktoriumi ir jo naudojimu vykdant užklausas.

3. Metodiniai nurodymai

Kokie yra užklausų tipai ir kokius duomenų apdorojimo būdus jos vykdo?

Duomenų bazei pateikiamos užklausos vykdo informacines paslaugas. Užklausos suteikia galimybes:

- ✚ Iš vienos arba kelių tarpusavyje susijusių lentelių išskirti reikalingus duomenis;
- ✚ Su duomenimis, pateiktais duomenų bazėje, atlikti skaičiavimus;
- ✚ Gauti pageidaujama duomenų lentelę;
- ✚ Lentelėse keisti duomenis – papildyti naujais įrašais ir šalinti nereikalingus.

Galima išskirti šiuos duomenų bazės užklausų tipus:

- ✚ Atrankos užklausos (*Select query*). Šios užklausos sudaromos norint pateikti laikiną informacijos darinį iš kelių susijusių lentelių ir kitų užklausų;

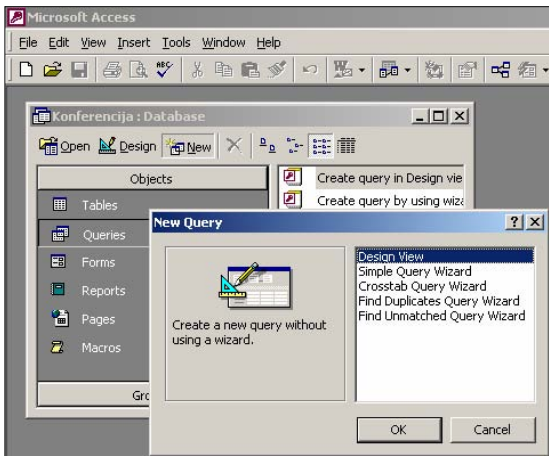
- ✚ Lentelių kūrimo užklauso (*Make-Table query*) sudaromos, norint pateikti naujas lenteles, kuriose būtų atrankos rezultatai;
- ✚ Duomenų tvarkymo užklauso. Jos keičia duomenų bazių lentelių duomenis.

Kaip atlikti dialoginę užklausa?

Kai kuriais atvejais norint patikslinti duomenis (pvz., parodant reikiamą informaciją apie autorių) ne visuomet naudinga iš naujo nustatyti paieškos kriterijus (pvz., nustatyti vis kitą pavardę, norint gauti informaciją apie kitą autorių). Naudingiau sukurti dialoginę (klausiančiąją) užklausa.

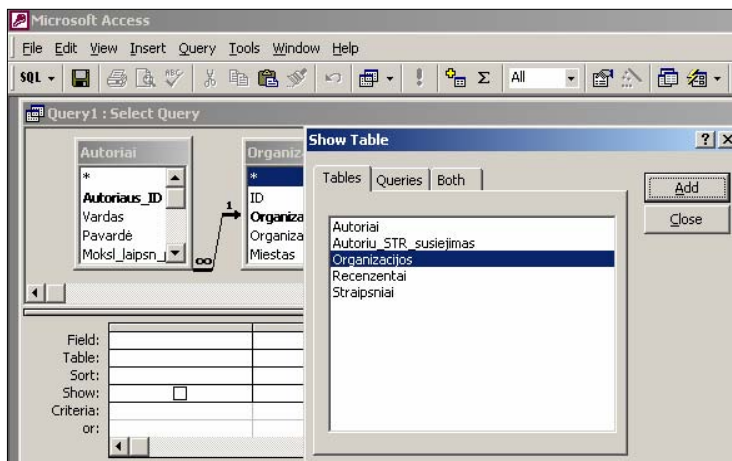
Norėdami išsiaiškinti straipsnio autoriaus adresą, užklausa formuosime iš dviejų lentelių: *Autoriai* ir *Organizacijos* duomenų.

Esant aktyviam duomenų bazės objektui *Query* spaudžiame priemonių juostos mygtuką *New* arba vykdome pagrindinio meniu komandą *Insert*→*Query* ir pasirenkame užklauso formavimo būdą *Desing View*.



3.50 pav. Užklauso formavimo būdo pasirinkimo pavyzdys

Dialogo lange *Show Table* pažymime lenteles, iš kurių duomenų formuosime užklausa, pavyzdžiui, *Autoriai* ir *Organizacijos* (žymėdami jų pavadinimus ir spausdami mygtuką *Add*).



3.51 pav. Lentelių parinkimo langas dialoginei užklausiai formuoti

Užklaustos projekto langas

Užklaustos laukai, kurių atitikmenys yra duomenų lentelėse, eilutėje *Show* pažymėti paukščiukais. Jie bus rodomi suformuotoje užklausoje.

Užklaustos laukas, kurio duomenys įvedami vykdant užklausą

3.52 pav. Dialoginės užklaustos projekto langas ir jos išsaugojimas

Klausiančioji užklausa, be lentelių laukų, į kuriuos duomenys surenkami iš atitinkamų lentelės stulpelių, turi papildomus laukus, skirtus įvesti duomenis kiekvieną kartą, kai vykdoma užklausa.

Papildomą lauką, skirtą sutapatinti duomenis, galima įvesti klaviatūra arba pasitelkiant reiškinių konstruktorių. Kaip naudoti reiškinių konstruktorių, aprašyta toliau pateiktuose šio darbo metodiniuose nurodymuose. Lauke *Field* reikėtų įvesti šiuos duomenis: [Autoriai] [Pavardė] *Like* [Įveskite pavardę]. Čia:

Autoriai – lentelės pavadinimas;

Pavardė – lauko lentelėje pavadinimas;

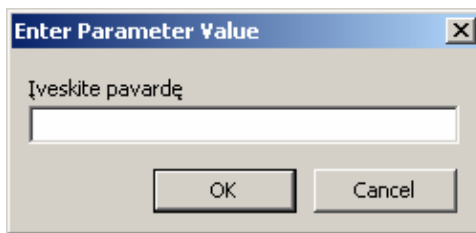
Like – tapatinimo operatorius, jis iš abiejų pusių atskiriamas tarpo ženklu;

[*Įveskite pavardę*] – dialoginės užklauskos pranešimo tekstas.

Papildomo lauko *Field* dalyje *Criteria* įvedama reikšmė –1, apibūdinanti lyginimo sąlygas: reikšmė –1 laukų sutapatinimo sąlygą tenkina, reikšmė 0 sąlygos netenkina. Žymuo *Show* išjungiamas, kai užklauskos rezultatui ši reikšmė nereikalinga.

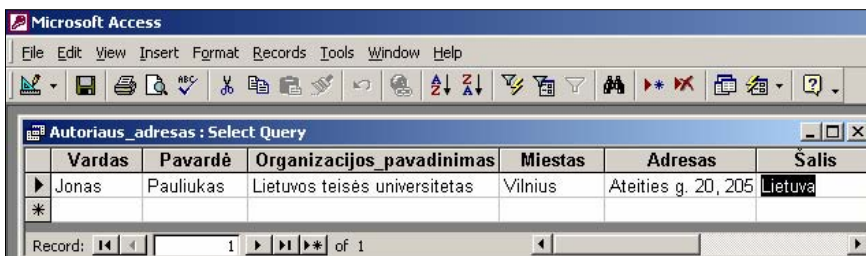
Sudarius užklauskos projektą, jį reikia išsaugoti kaip užklausa, įvedant užklauskos pavadinimą, pavyzdžiui, *Autoriaus_adresas*.

Vykdamt dialoginę užklausa (užklausa vykdoma dukart spaudžiant kairįjį pelės klavišą ties užklauskos pavadinimu) pirmiausiai pasirodys pranešimas, reikės įvesti lauko reikšmę (3.53 pav.). Ją įvedus programa analizuos duomenis ir juos pateiks pagal užklauskos sąlygas.



3.53 pav. Dialoginės užklauskos langas pavardei įvesti

Suformavus užklausa *Autoriaus_adresas* įvedama pavardė, pavyzdžiui, Pauliukas, ir toliau pateiktoje lentelėje pagal anksčiau sudarytą užklauskos projektą gaunamas užklauskos rezultatas.



3.54 pav. Dialoginės užklauskos pateikti duomenys įvedus pavardę Pauliukas

Kaip naudotis reiškinių konstruktoriumi sudarant užklauską?

Skaičiuojamieji laukai – vieni svarbiausių užklauskų ir atskaitų struktūrinių elementų. Juose aprašoma, kaip iš bazėje saugomų duomenų gaunami vartotojui reikalingi duomenys. Duomenims laikyti skirtose lentelėse tokių laukų nėra.

Reiškinių konstruktorius skirtas sudaryti sudėtingesnėms duomenų atrankos sąlygoms.

Reiškinius į skaičiuojamus laukus galima įvesti klaviatūra arba vykdant specialią komandą, vadinamą reiškinių konstruktoriumi (*Builder*). Esant užklauskos projekto lange, priemonių juostos mygtuku *Build* atveriamas reiškinių konstruktoriaus langas (*Expression Builder*).

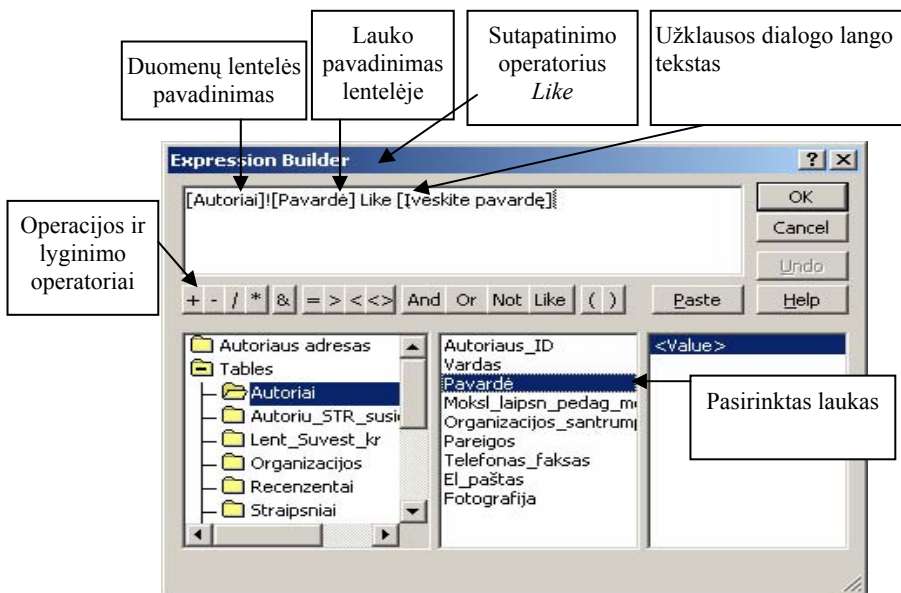
Mygtukas *Build* skirtas reiškinių konstruktoriaus langui iškviesti



3.55 pav. Bendroji priemonių juosta

Kitas būdas iškviesti reiškinių konstruktorių – esant užklauskos projekto lange ties lauku *Field* spausti dešinįjį pelės klavišą ir pasirodžius kontekstiniam menui vykdyti komandą *Build*. Ši komanda atveria reiškinių konstruktoriaus langą *Expression Builder*. Reiškinių konstruktoriaus lango viršuje sudaromas reiškinių laukas, kurio elementai parenkami iš lango apačioje matomų sąrašų ir iš lango viduryje matomų dažniausiai naudojamų lyginimo operatorių sąrašo.

Reiškinių konstruktoriaus lango kairėje apačioje matomi sąrašai – tai duomenų bazės objektai: lentelės, užklausos, funkcijos, konstantos ir kita. Kiekvienas šis objektas gali turėti jam būdingas grupes arba kategorijas – tai vidurinioji reiškinių konstruktoriaus skiltis. Kategorijos ir grupės gali turėti reikšmes arba elementus – skiltis dešinėje.



3.56 pav. Reiškinių konstruktoriaus langas ir priemonės, naudojamos sudarant užklausą

Atliekant anksčiau aprašytą užduoties pavyzdį pavardės, esančios *Aukorių* lentelėje, lyginamos su užklausos pranešimo langelyje įvesta pavarde. Sudarant užklausos projektą projekto lango eilutėje *Field* iškviečiamas reiškinių konstruktorius.

Reiškinių konstruktoriaus lange (*Expression Builder*) pasirenkama lentelė *Aukoriai* ir dvigubu kairiojo pelės klavišo paspaudimu laukas *Pavardė* įrašomas į sudaromą reiškinių lauką. Konstruktoriaus lango viršuje pelės klavišu spaudžiant operatoriaus žymą *Like* į sudaromą reiškinių įvedamas duomenų sutapatavimo operatorius (*Like*) ir klaviatūra laužtiniuose skliaustuose renkamas užklausos pranešimo tekstas [Įveskite pavardę].

Kontroliniai klausimai

1. Kokius žinote skirtingos paskirties užklausų tipus?
2. Kokia užklausos veiksmų kūrimo seka?
3. Kam skirtos užklausos projekto lango eilutės *Field, Criteria, Show*?
4. Kaip užklausos rezultatus išdėstyti didėjimo arba mažėjimo tvarka?
5. Ar galima išsaugoti suformuotą užklausa?
6. Kaip vykdomos užklausos?
7. Kada patogu naudoti dialogines užklausas?
8. Kam skirtas reiškinių konstruktorius?
9. Kaip reiškinių konstruktoriumi parinkti argumentus ir funkcijas?

3.7.6. Suvestinių duomenų užklausų atlikimo būdai

1. Darbo tikslas

Išmokti duomenų bazėje atlikti suvestines užklausas.

2. Darbo užduotys

1. Iškviškite programą *MS Access*. Atverkite sukurtą duomenų bazę.
2. Išsiaiškinkite, kokio tipo užklausos vadinamos suvestinėmis užklausomis.
3. Atlikite suvestinę užklausa, pavyzdžiui, formuodami ją iš lentelės *Autoriai*. Užklausos tikslas – pateikti autorių, parašiusių konferencijai skirtus straipsnius, skaičių pagal pareigas skirtingose organizacijose.
4. Susipažinkite su suvestinės užklausos projekto langu bei projekto sudarymo būdu.
5. Susipažinkite su reiškinių konstruktoriumi ir jo naudojimu vykdam užklausas.

3. Metodiniai nurodymai

Kam naudojamos suvestinės užklauskos ir kokius duomenų apdorojimo būdus jos atlieka?

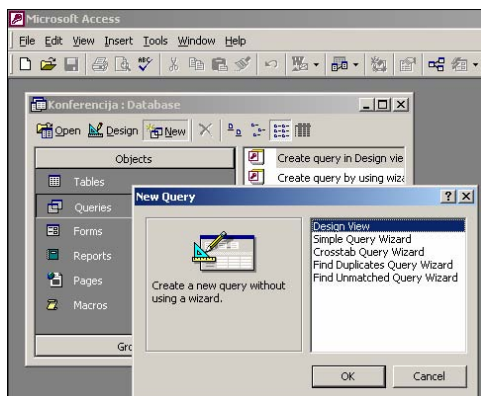
Duomenų bazėse laikomi duomenys dažnai naudojami suvestinėms sudaryti. Suvestinėmis vadinami dokumentai, kuriuose yra apibendrinti duomenys, t. y. pagal įvairius duomenų požymius susumuoti rezultatai, statistiniai duomenų įverčiai ir panašiai.

Sudarant suvestines duomenų užklauskas naudojamos duomenų skaičiavimo funkcijos, *MS Access* programoje vadinamos agregatinėmis (*Aggregate*). Šių funkcijų argumentai yra duomenų bazės duomenys. Dažniausiai šios funkcijos naudojamos ataskaitoms sudaryti, tačiau jas galima taikyti ir atliekant užklauskas.

Kaip iškviešti projekto langą suvestinei užklausiai formuoti?

Užklausą formuosime iš lentelės *Autoriai* duomenų.

Esant aktyviam duomenų bazės objektui *Query* spaudžiame priemonių juostos mygtuką *New* arba vykdome pagrindinio meniu komandą *Insert*→*Query* ir pasirenkame užklauskos formavimo būdą *Design View*.



3.57 pav. Užklauskos formavimo būdo *Design View* pasirinkimas

Dialogo lange *Show Table* spausdami mygtuką *Add* išrenkame lentelės, iš kurių duomenų formuosime užklausą, pavyzdžiui, lentelę *Autoriai*.

Kaip sudaryti suvestinės užklausoje projektą?

Panagrinėsime pavyzdį, kaip sudaryti suvestinį užklausoje projektą, kuris pateiktų organizacijų autorių skaičių pagal dalyvaujančių konferencijoje pareigas.

Suvestinės užklausoje projektas sudaromas šia seka:

- ✚ Aprašomi suvestinės užklausoje laukai;
- ✚ Vykdoma pagrindinio meniu komanda *View*→*Totals*;
- ✚ Parenkamos tinkamos suvestinės užklausoje laukų funkcijos.

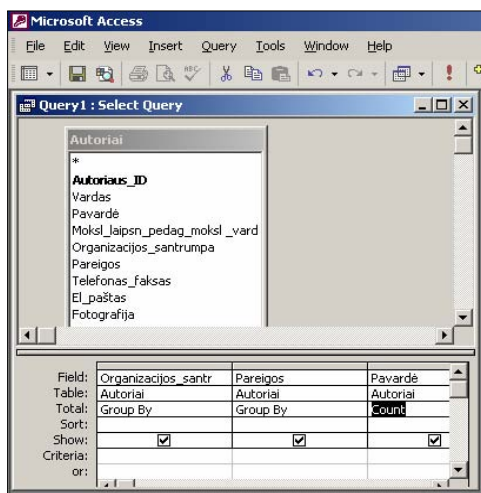
Suvestinės užklausoje laukų aprašymas

Projektuojamoje užklausoje reikalingi du grupavimo požymiai, esantys lentelės *Autoriai* laukuose *Organizacijos santrumpa* ir *Pareigos*. Iš lentelės *Autoriai* į projekto eilutę *Field* pele nutempiami laukų *Organizacijos santrumpa* ir *Pareigos* pavadinimai.

Projekte nurodant grupavimo laukus formuojami skaičiavimo argumentų laukai. Bus skaičiuojamas pavardžių skaičius, todėl į trečiąjį projekto stulpelį nutempiamas laukas *Pavardė*.

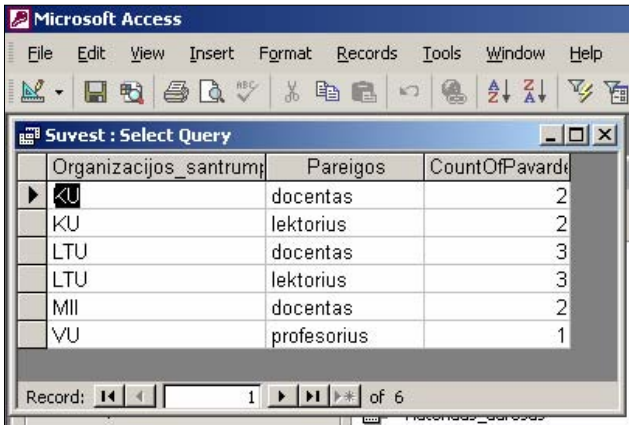
Įvykdžius komandą *View*→*Totals* projekto lange įterpiama eilutė *Total*, parenkamos tinkamos suvestinės užklausoje laukų funkcijos.

Total eilutėje organizacijos santrumpai ir pareigoms įrašyti paliekami reikšmės grupavimo požymiai ir parenkama skaičiuojamo *Pavardės* lauko įrašų skaičiavimo funkcija *Count*.



3.58 pav. Suvestinės užklausoje projekto langas

Suvestinės užklauskos rezultatas naudojant lentelės *Autoriai* duomenis pateikiamas 3.59 pav.



	Organizacijos_santrump	Pareigos	CountOfPavard
▶	KU	docentas	2
	KU	lektorius	2
	LTU	docentas	3
	LTU	lektorius	3
	MII	docentas	2
	VU	profesorius	1

3.59 pav. Suvestinės užklauskos rezultatai

Kontroliniai klausimai

1. Kokiems duomenims pateikti naudojamos suvestinės užklauskos?
2. Kokia suvestinės užklauskos veiksmų kūrimo seka?
3. Kam skirta eilutė *Total* suvestinės užklauskos projekto lange?
4. Ar galima išsaugoti suformuotą suvestinę užklauską?
5. Kaip vykdomos suvestinės užklauskos?

3.7.7. Kelias lenteles siejančių užklausų atlikimo būdai

1. Darbo tikslas

Išmokti atlikti užklausą susiejant kelių lentelių duomenis. Atliekant šio tipo užklausą duomenys iš duomenų bazės atrenkami pagal vartotojo sudarytą projektą.

2. Darbo užduotys

1. Iškvieskite programą *MS Access*. Atverkite sukurtą duomenų bazę.
2. Išsiaiškinkite kelias lenteles siejančių užklausų atlikimo būdus ir šių užklausų naudojimo atvejus.
3. Suprojektuokite užklausą, pavyzdžiui, formuodami ją iš lentelių *Autoriai*, *Straipsniai* ir *Autorių STR_susiejimas*. Užklaustos tikslas – išsiaiškinti, kokius straipsnius ir kokia tema pateikė autoriai.
4. Susipažinkite su užklaustos projekto langu bei projekto sudarymo priemonėmis.
5. Sudarykite užklausą, kuri pagal temas išrinktų autorius ir jų straipsnius.

3. Metodiniai nurodymai

Kokiais atvejais naudojamos kelias lenteles siejančios užklaustos?

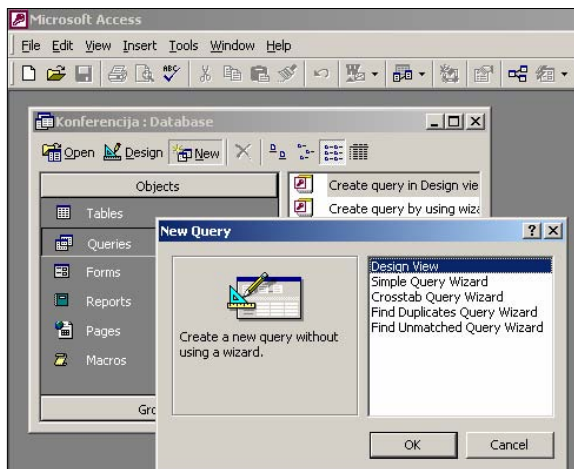
Užklausa gali būti sudaroma susiejant kelias lenteles ir iš šių tarpusavyje susijusių lentelių išskiriant reikalingus duomenis. Šio tipo užklaustos priskiriamos prie atrankos užklausų (*Select query*). Jos sudaromos norint pateikti laikiną informacijos darinį iš tarpusavyje susijusių kelių lentelių ir jau suformuotų užklausų.

Kaip sukurti kelių lentelių užklausą?

Kai kuriais atvejais norint pateikti duomenis (pvz., kai parodoma reikiama informacija apie straipsnį ir jo autorių kolektyvą) juos reikia išrinkti iš kelių lentelių.

Užklausą atliksime susiedami tris lenteles: *Autoriai*, *Straipsniai* ir *Autorių STR_susiejimas*.

Esant aktyviam duomenų bazės objektui *Query* spaudžiamas priemonių juostos mygtukas *New* arba vykdoma pagrindinio meniu komanda *Insert*→*Query* ir pasirenkama užklauso formavimo būdas *Design View*.



3.60 pav. Užklauso formavimo būdo *Design View* pasirinkimas

Užklauso projekto langas

Dialogo lange *Show Table* išrenkamos lentelės, iš kurių duomenų bus formuojama užklausa, pavyzdžiui, *Autoriai*, *Straipsniai* ir *Autorių STR_susiejimas* (žymint jų pavadinimus ir spaudžiant mygtuką *Add*).

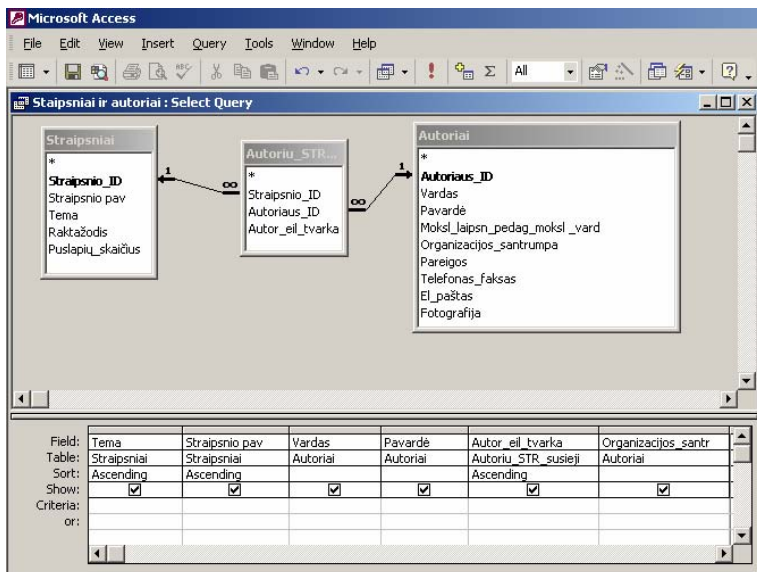
Lentelių ryšių projekte nurodyti ryšiai bus pažymėti ir užklauso projekto lange.

Pasirenkami laukai iš atskirų lentelių (3.61 pav.).

Sudarius užklauso projektą ir uždarant projekto langą, programa paprašys jį išsaugoti kaip užklausą. Tuomet reikia įvesti užklauso pavadinimą, pavyzdžiui, *Straipsniai ir autoriai*, bei jį išsaugoti.

Vykdamt jau suformuotą užklausą duomenys bus pateikti pagal sudarytą užklauso projektą.

Įvykdžius užklausą *Straipsniai ir autoriai*, gaunamas užklauso rezultatas (3.62 pav.).



3.61 pav. Kelių lentelių užklauso projekto langas

Tema	Straipsnio pav	Vardas	Pavardė	Autor_eil_tvar	Organizacijos
Informacinės sistemos	Ekspertinių žinių metodai sprendimų priėmimo sistemose	Dalia	Ivanauskienė	1	Mil
Informacinės sistemos	Informacinių sistemų projektavimas	Laimonas	Milenas	1	KU
Informacinės sistemos	Informacinių sistemų projektavimas	Dalia	Ivanauskienė	2	Mil
Informacinės sistemos	Šablono metodo taikymas kuriant sistemas	Rita	Dapkūnaite	1	Mil
Informacinės technologijos	Baudžiamosios teisės sankcijų informacinis įvertinimas	Yladišlovas	Poška	1	LTU
Informacinės technologijos	Baudžiamosios teisės sankcijų informacinis įvertinimas	Mykolas	Baniulis	2	LTU
Informacinės technologijos	Baudžiamosios teisės sankcijų informacinis įvertinimas	Eglė	Polikaitė	3	LTU
Informacinės technologijos	Kompiuterinių nusikaltimų tyrimo problemos	Ytėtautas	Jonaitis	1	LTU
Informacinės technologijos	Kompiuterinių nusikaltimų tyrimo problemos	Simonas	Laukus	2	LTU
Studijos	Informacinės technologijos mokykloje	Gintautas	Varkalys	1	KU
Studijos	Informacinės technologijos mokykloje	Rūta	Milūtė	2	KU
Studijos	Informacinės technologijos mokykloje	Gediminas	Dirgela	3	KU
Studijos	Informacinių technologijų taikymas mokymo procese	Jonas	Pauliukas	1	LTU
Studijos	Nuotolinio mokymo elementai	Vitalija	Miniotienė	1	VU

3.62 pav. Kelių lentelių užklauso rezultatai

Kontroliniai klausimai

1. Kokius žinote skirtingos paskirties užklausų tipus?
2. Kokia yra kelias lenteles siejančios užklausos veiksmų kūrimo seka?
3. Kam skirtos užklausos projekto lango eilutės *Field*, *Criteria*, *Show*?
4. Kaip užklausos rezultatus išdėstyti didėjimo arba mažėjimo tvarka?
5. Kaip redaguoti suformuotą užklausa?

3.7.8. Kryžminių suvestinių duomenų užklausos

1. Darbo tikslas

Išmokti duomenų bazėje atlikti kryžmines suvestines užklausas.

2. Darbo užduotys

1. Iškviestite programą *MS Access*. Atverkite sukurtą duomenų bazę.
2. Išsiaiškinkite, kokio tipo užklausos vadinamos kryžminėmis suvestinėmis užklausomis.
3. Suprojektuokite kryžminę suvestinę užklausa, kurios tikslas – suskaičiuoti, kiek skirtingų organizacijų autorių pateikė straipsnius skirtingomis temomis, formuodami ją iš užklausos *Straipsniai ir autoriai*.
4. Susipažinkite su kryžminės suvestinės užklausos formavimu pasitelkiant vedlį. Atlikite kryžminės suvestinės užklausos formavimo etapus:
 - Vedlio lango iškvietimas kryžminei suvestinei užklausiai formuoti;
 - Kryžminei suvestinei užklausiai reikalingo duomenų šaltinio (lentelės, užklausos ir pan.) parinkimas;
 - Kryžminei suvestinei užklausiai reikalingų eilučių požymių laukų parinkimas;
 - Kryžminei suvestinei užklausiai reikalingo stulpelio požymio lauko parinkimas;

- ✚ Kryžminės suvestinės užklauso lauko, skirto skaičiavimams atlikti, parinkimas;
- ✚ Kryžminės suvestinės užklauso vykdymas ir išsaugojimas.

3. Metodiniai nurodymai

Kam naudojamos kryžminės suvestinės užklauso?

Kryžminės suvestinės užklauso plačiai taikomos analizuojant duomenis, duomenų pasiskirstymą pagal du požymius. Kryžminių suvestinių užklauso eilutės skiriamos vieno požymio reikšmėms, o stulpeliai – kito požymio reikšmėms pateikti. Pasitelkiant kryžmines suvestines užklauso gali būti skaičiuojamas įrašų kiekis, laukų reikšmių sumos, statistikos duomenys.

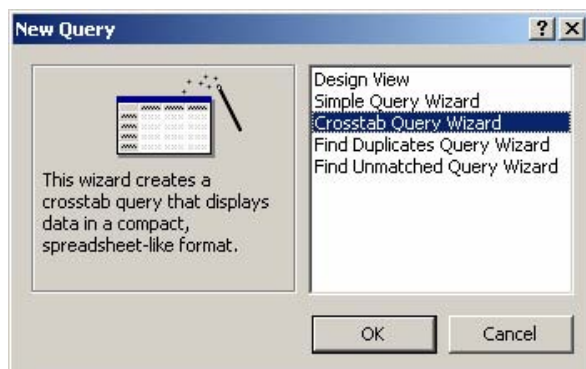
Sudarant kryžmines suvestines užklauso duomenų bazėse naudojamos specialios *Crosstab Query* tipo užklauso.

Kaip iškviešti vedlio langą kryžminei suvestinei užklausiai formuoti?

Užklausa bus formuojama iš užklauso *Straipsniai ir Autoriai* (žr. 3.7.7. praktinį darbą) duomenų.

Esant aktyviam duomenų bazės objektui *Query* spaudžiamas priemonių juostos mygtukas *New* arba vykdoma pagrindinio meniu komanda *Insert*→*Query* bei pasirenkamas užklauso formavimo būdas *Crosstab Query Wizard*.

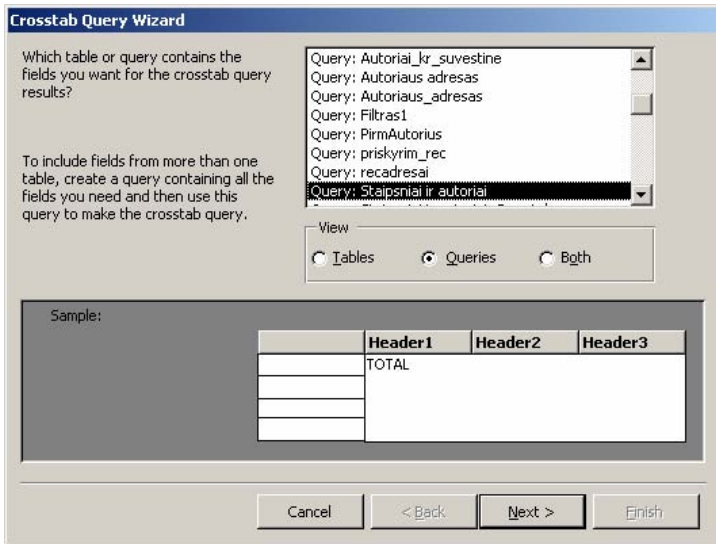
Šias užklauso patogu rengti naudojantis vedliu (*Wizard*).



3.63 pav. Kryžminės užklauso formavimo būdo pasirinkimas vykdant komandą *Insert*→*Query*

Kaip parinkti kryžminei suvestinei užklausiai reikalingą duomenų šaltinį?


Įvykdžius komandą *Insert*→*Query* atveriamas *Crosstab Query Wizard* langas, kuriame reikia parinkti duomenų šaltinį, pavyzdžiui, užklausą *Straipsniai ir autoriai*.



3.64 pav. Kryžminės užklaustos duomenų šaltinio parinkimas

Parinkus šaltinį – užklausą *Straipsniai ir autoriai* – ir paspaudus mygtuką *Next*, bus atvertas langas eilutėms skirto požymio laukui parinkti.

Kaip parinkti kryžminei suvestinei užklausiai atlikti reikalingus eilučių požymių laukus?

Iš sąrašo *Available Fields* mygtuku  perkeliame eilutėms skirto požymio lauko pavadinimą *Organizacijos santrumpa* į sąrašą *Selected Fields*. Sudarant kryžmines suvestines užklausas suvestinių eilutėse gali būti keli skirtingi laukai, tuomet duomenys bus paskirstomi pagal kelis požymius.

Mygtuku *Next* pereinama į stulpelių požymio parinkimo langą.

Crosstab Query Wizard

Which Fields' values do you want as row headings?

You can select up to three fields.

Select Fields in the order you want information sorted. For example, you could sort and group values by Country and then Region.

Available Fields:

- Tema
- Straipsnio pav
- Vardas
- Pavardė
- Autor_eil_tvarka

Selected Fields:

- Organizacijos_santrumpa

Sample:

Organizacijos	Header1	Header2	Header3
Organizacijos_s	TOTAL		
Organizacijos_s			
Organizacijos_s			
Organizacijos_s			

Buttons: Cancel, < Back, Next >, Finish

3.65 pav. Eilutės požymio laukų parinkimas

Kaip parinkti kryžminei suvestinei užklausiai atlikti reikalingą stulpelio požymio lauką?

Šis langas yra analogiškas eilučių parinkimo langui, tačiau čia galima parinkti tik vieną lauko pavadinimą. Stulpelio pavadinimui kurti rinksimės *Temos* lauką.

Crosstab Query Wizard

Which field's values do you want as column headings?

For example, you would select Employee Name to see each employee's name as a column heading.

Available Fields:

- Autoriaus_ID
- Vardas
- Pavardė
- Moksl_laipsn_pedag_moksl_vard
- Pareigos
- Telefonas_faksas
- El_paštas
- Darys_pranešimą
- Straipsnio_ID
- Tema

Sample:

Organizacijos	Tema1	Tema2	Tema3
Organizacijos_s	TOTAL		
Organizacijos_s			
Organizacijos_s			
Organizacijos_s			

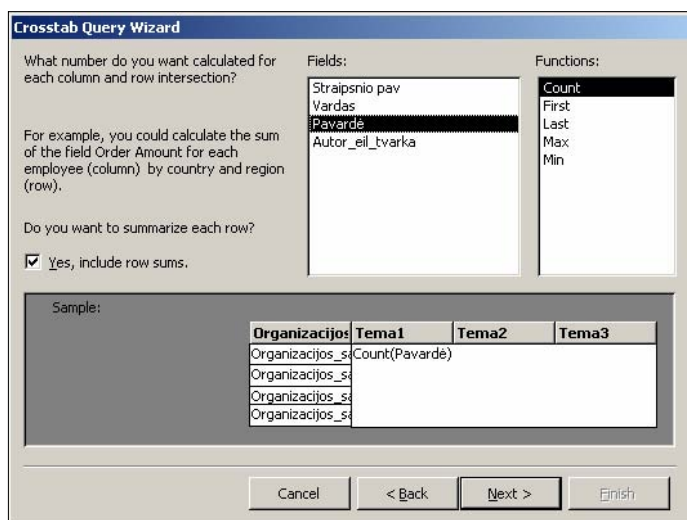
Buttons: Cancel, < Back, Next >, Finish

3.66 pav. Stulpelio požymio laukų parinkimas

Kaip parinkti kryžminės suvestinės užklauso lauką, skirtą skaičiavimams atlikti?

Spaudžiant mygtuką *Next* atveriamas dar vienas langas, kuriame reikia nurodyti lauko pavadinimą, skirtą skaičiavimams atlikti. Vykdam kryžminę suvestinę užklausa bus skaičiuojamas autorių kiekis, todėl reikia pažymėti lauko reikšmę *Pavardė*, o funkcijų grupėje – įrašams skaičiuoti skirtą funkciją *Count*.

Sąrašė *Functions* funkcijų rinkinys priklauso nuo pasirinkto lauko duomenų tipo. Esant skaitinėms ir tekstinėms reikšmėms pateikiamas skirtingas funkcijų sąrašas. Mūsų pateiktame pavyzdyje matyti funkcijų sąrašas, skirtas dirbti su teksto lauku.

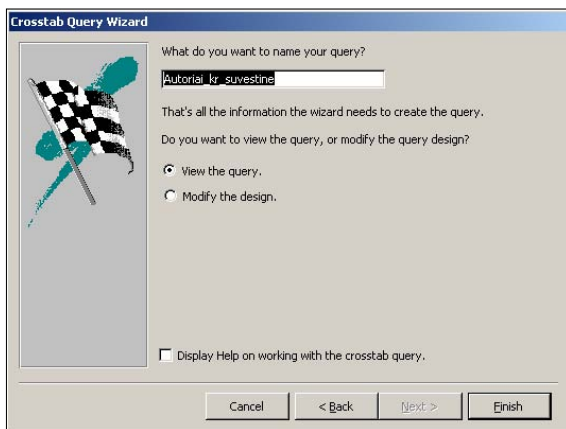


3.67 pav. Lauko, skirto skaičiavimams kryžminėje užklausoje atlikti, parinkimas

Esant pažymėtam jungikliui *Yes, Include row sums* turi būti skaičiuojamos kiekvienos eilutės duomenų sumos.

Kaip išsaugoti ir vykdyti kryžminę suvestinę užklausa?

Paskutiniame kryžminės suvestinės užklauso lange nurodomas užklauso pavadinimas, ji formuojama mygtuku *Finish*.



3.68 pav. Kryžminės užklauskos išsaugojimo ir vykdymo langas

Darbo rezultatas – kryžminė suvestinė duomenų lentelė, kurioje matyti autorių iš skirtingų organizacijų ir tų pačių organizacijų autorių, rašančių viena tematika, skaičius.

Organizacijos	Total Of Pavardė	Informacinės sistemos	Informacinės technologijos	Studijos
KU	4	1		3
LTU	6		5	1
MII	3	3		
VU	1			1

3.69 pav. Kryžminės užklauskos rezultatų lentelė

Kontroliniai klausimai

1. Kokiems duomenims pateikti naudojamos kryžminės suvestinės užklauskos?
2. Kokia yra kryžminės suvestinės užklauskos veiksmų kūrimo seka?
3. Kokie požymių laukai parenkami kryžminėje suvestinėje užklausoje?
4. Ar galima išsaugoti suformuotą kryžminę suvestinę užklauską?
5. Kaip vykdomos kryžminės suvestinės užklauskos?

3.7.9. Lentelių kūrimo, papildymo, įrašų šalinimo, duomenų keitimo užklausa

1. Darbo tikslas

Išmokti duomenų bazėje formuoti išvestinę lentelę, sukurti užklausa, leidžiančias pildyti arba šalinti įrašus ir pakeisti lauko lentelės reikšmes.

2. Darbo užduotys

1. Iškviestite programą *MS Access*. Atverkite sukurtą duomenų bazę.
2. Išsiaiškinkite, kokio tipo užklausa vadinamos lentelių kūrimo ir prijungimo užklausomis.
3. Atrankos užklausiai suteikite lentelės kūrimo funkciją, pavyzdžiui, užklausiai *Suvestinė* suteikite lentelių kūrimo funkciją, kad ši užklausa formuotų duomenų bazės lentelę, kurios pavadinimas – *Suvestinė*.
4. Panaikinkite sukurtosios užklausa lentelės kūrimo funkciją.
5. Išsiaiškinkite, kaip sudaryti lentelių papildymo užklausa?
6. Išsiaiškinkite, kaip sudaromos įrašų šalinimo iš lentelės užklausa, sukurkite įrašų šalinimo užklausa.
7. Išsiaiškinkite, kaip organizuojamos lentelės duomenų keitimo užklausa, sukurkite duomenų keitimo užklausa.

3. Metodiniai nurodymai

Kokiu tikslu duomenų bazėse naudojamos lentelių kūrimo ir prijungimo užklausa?

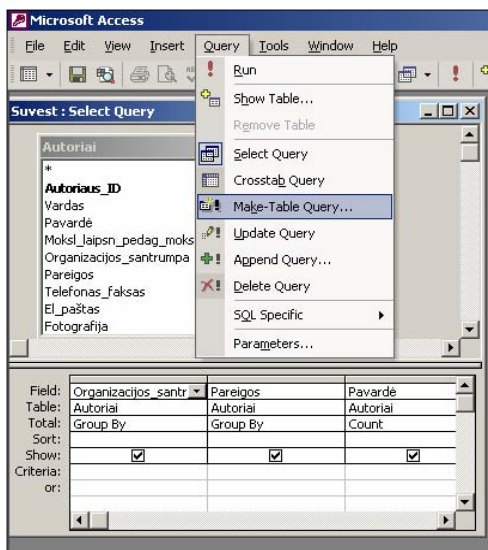
Visos anksčiau nagrinėtos užklausa vadinamos atrankos užklausomis, nes duomenis iš duomenų bazės jos atrenka pagal vartotojo nurodytas sąlygas. Tokių užklausa lentelės formuojamos kas kart, kai tik vykdoma užklausa, patys užklausa lentelės duomenys nėra saugomi.

Jeigu atrinkimo užklausa rezultatus reikia išsaugoti kaip lentelę ir laikyti su kitais duomenų bazės duomenimis, atrankos užklausa reikia pertvarkyti į lentelės kūrimo užklausa.

Kaip pertvarkyti užklausą, kad ji virstų nauja duomenų bazės lentele?

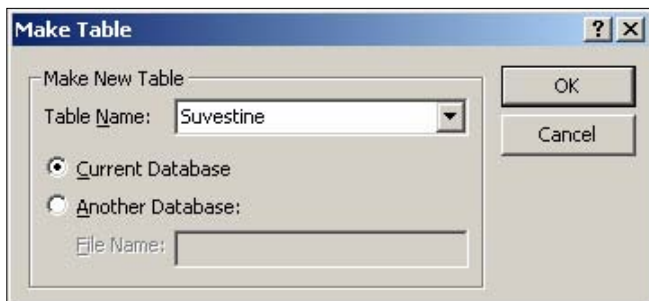
Duomenų bazę papildysime nauja suvestinės užklaustos lentele. Jeigu anksčiau sudarytą suvestinių užklausą reikia išsaugoti kaip naują lentelę, atliekami šie veiksmai:

- ✚ Pagrindinio lango užklausų objekte pažymimas užklaustos pavadinimas *Suvestinė*;
- ✚ Nuspaudžiamas mygtukas *Design* ir atveriamas užklaustos projekto langas;
- ✚ Vykdoma pagrindinio meniu komanda *Query*→*Make Table Query*.

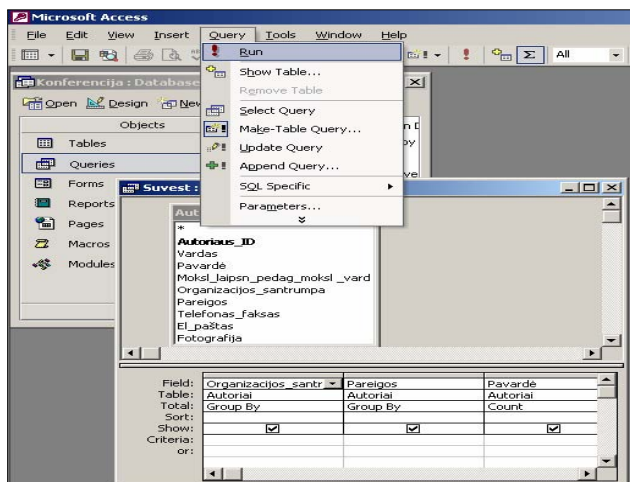


3.70 pav. Lentelės kūrimo užklaustos langas

Įvykdžius komandą *Query*→*Make Table Query* pateikiamas langas *Make Table*, skirtas naujai kuriamos lentelės pavadinimui, pavyzdžiui, *Suvestinė*, įvesti. Atlikus nurodytus veiksmus suvestinės atrankos užklausiai suteikiama lentelės kūrimo funkcija. Lentelė sukuriamą, priemonių juostos mygtuku *Run* arba komanda *Query*→*Run*. Vykdamas lentelių kūrimo užklausą išvestinė duomenų lentelė kiekvieną kartą perrašoma iš naujo.

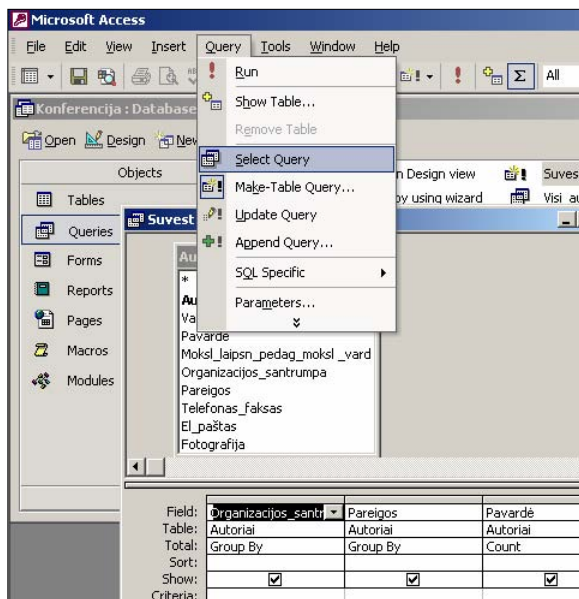


3.71 pav. Kuriamos lentelės pavadinimo įrašymo langas

3.72 pav. Lentelės kūrimo komandos
Query→*Run* vykdymas

Kaip atšaukti lentelės kūrimo funkcijas?

Suvestinės užklauskos projekto lange vykdant komandą *Query*→*Select Query* galima panaikinti lentelės kūrimo ir palikti tik atrankos funkciją, t. y. grįžti prie buvusių užklauskos funkcijų, pavyzdžiui, *Suvestinės* užklauskos.



3.73 pav. Lentelės kūrimo iš užklauskos funkcijos panaikinimo komanda

Kaip sudaryti lentelių papildymo užklauskas?

Kai duomenis reikia papildyti, užklauską patogų pertvarkyti į lentelės papildymo užklauską, kuri atrenkamus įrašus prijungtų prie lentelėje esančių įrašų. Vykdamt prijungimo užklauską nekuriama nauja lentelė, todėl iš pradžių reikia sukurti užklauskos rezultatams pritaikytą tuščią lentelę.

Lentelių papildymo užklausa vykdoma komanda *Query*→*Append Query*.

Pavyzdžiui, lentelės papildymo užklausa galėtų būti vykdoma į suvestinių lentelę įrašant metus, o vėliau, kitais metais, rengiant naujas konferencijas, šiuos duomenis papildant naujais.

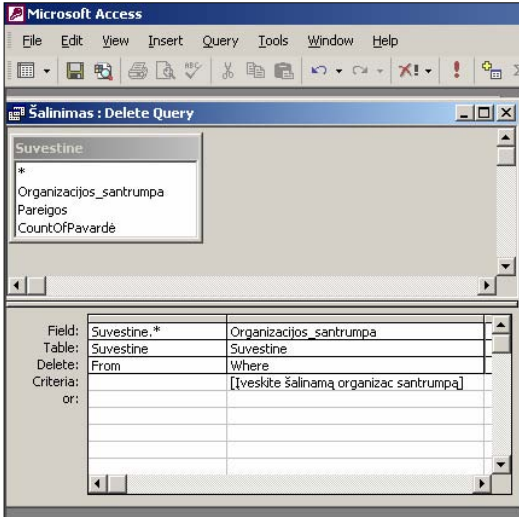
Kaip sudaromos duomenų šalinimo užklauskos?

Tarkime, iš suvestinės lentelės reikia pašalinti įrašus, kurių organizacijos santrumpa – VU.

Sudarant šalinimo užklauską užklauskos projekto lange reikia pasirinkti lentelę *Suvestinė* ir vykdyti komandą *Query*→*Delete Query*. Ši komanda suformuoja projekto eilutę *Delete*.

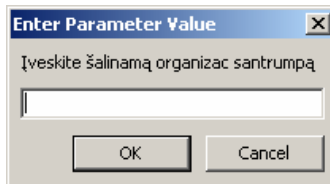
Į pirmąjį stulpelio lauką perkelsime simbolį *, nurodanti, kad į šalinimo užklausa turi būti įtraukiami visi lentelės laukai. Žyma *From* nurodo, kad iš šios lentelės bus šalinami įrašai. Žyma *Where* informuoja, kad tai – šalinamų įrašų požymių aprašymo stulpelis.

Toliau pateiktame lange parodytas dialoginės užklaustos lango projektas suvestinės lentelei sudaryti.



3.74 pav. Šalinimo įrašų užklaustos projekto langas

Baigta formuoti ši lentelė vykdant užklausa pateikia vartotojui dialogo langą, skirtą šalinamo įrašo organizacijos santrumpai įrašyti.



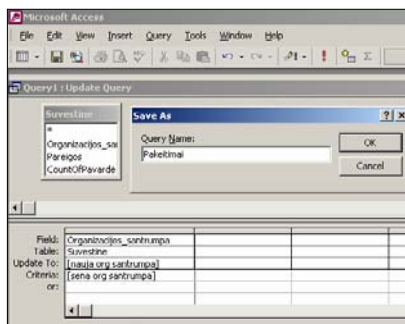
3.75 pav. Dialogo langas šalinamų įrašų organizacijos santrumpai įvesti

Panašiai lentelėse parengiamos ir kaskadinio šalinimo užklaustos, atrenkančios duomenis pagal ryšius *vienas su daugeliu*. Kad užklausa vyktų sėkmingai, reikia, jog būtų įjungta ryšio savybė *Cascade Delete Related Records*.

Kaip atliekamos duomenų keitimo užklausos?

Tarkime, reikia organizuoti dialoginę užklausą, leidžiančią suvestinių lentelėje pakeisti organizacijos santrumpą. Tuomet reikia:

- ✚ Sudaryti duomenų keitimo užklausos projekto langą ir įkelti į jį lentelės, kurios duomenis keisime, pavadinimą;
- ✚ Vykdyti pagrindinio meniu komandą *Query*→*Update Query*;
- ✚ Į *Field* lauką perkelti keičiamo lauko pavadinimą;
- ✚ Eilutėje *Update to* lauztiniuose skliaustuose įrašyti pranešimą, kuris bus pateikiamas įvedant naują lauko reikšmę;
- ✚ Eilutėje *Criteria* įrašyti pranešimą, kuris bus pateikiamas įvedant naują lauko reikšmę;
- ✚ Išsaugoti pakeitimo užklausą arba ją iškart vykdyti.



3.76 pav. Duomenų keitimo užklausos formavimo langas vykdant komandą *Query*→*Update Query* ir duomenų keitimo užklausos išsaugojimas

Vykdamas keitimo užklausą programa pateikia du dialogo langus naujai ir senai reikšmėms įrašyti.

Kontroliniai klausimai

1. Nurodykite veiksmus, reikalingus atliekant lentelių kūrimo užklausą.
2. Paašškinkite, kaip galima sukurti užklausą, skirtą papildyti lentelę naujais duomenimis.
3. Kokia užklausos įrašų šalinimo iš lentelės veiksmų kūrimo seka?
4. Kokia užklausos lauko reikšmių keitimo lentelėse veiksmų kūrimo seka?
5. Kaip vykdomos sukurtos užklausos?

3.7.10. Ataskaitos duomenų bazėje

1. Darbo tikslas

Išsiaiškinti, kam skirtos ataskaitos, kokiems duomenų apdorojimo būdams vykdyti reikalingos ataskaitos. Išmokti duomenų bazėje, sukurtoje programa *Microsoft Access*, sudaryti ir išspausdinti ataskaitą.

2. Darbo užduotys

1. Iškviestite programą *MS Access*. Atverkite sukurtą duomenų bazę.
2. Išsiaiškinkite, kam skirtos ataskaitos ir kaip jos apdoroja duomenis.
3. Sudarykite ataskaitą, pavyzdžiui, formuodami ją iš užklauskos *Straipsniai ir autoriai* pagal konferencijoje dalyvaujančių autorių sąrašą, sugrupuokite juos pagal temas, skaitomus pranešimus ir autorių eilės tvarką.
4. Susipažinkite, kaip ataskaita formuojama pasitelkiant vedlį.
5. Išspausdinkite ataskaitą.

3. Metodiniai nurodymai

Kam naudojamos ataskaitos (Reports) ir kokiems duomenų apdorojimo būdams vykdyti jos skirtos?

Duomenų bazių ataskaitos padeda parengti patogius duomenims analizuoti ir gražios išvaizdos dokumentus bei juos spausdina. Ataskaitų duomenų šaltiniai gali būti duomenų bazių lentelės ir užklauskos. Lentelių bei užklauskų duomenys taip pat gali būti spausdinami nerengiant ataskaitų, tačiau taip išspausdinti duomenys dažnai netenkina dokumentams keliamų reikalavimų – yra neišvaizdūs, juose nėra patogų peržiūrėti ir analizuoti duomenis.

Projektuojant ataskaitas duomenų šaltiniai pasirenkami iš vienos arba kelių tarpusavyje susijusių lentelių, tuo pat metu galima naudotis ir užklausoje išskirtais duomenimis. Ataskaitoms projektuoti naudojamas jų projekto langas arba ataskaitų kūrimo vedliai, pateikiantys šablonus, kuriuos vartotojas gali pritaikyti kurdamas savo ataskaitą ir įvesdamas į juos reikiamus lentelių ir užklauskų laukus.

Ataskaitose duomenys grupuojami ir rikiuojami pagal vartotojo parinktus duomenis, į jas galima įterpti grupavimo požymius, skaičiavimo laukus ir susumuoti duomenis.

Rengiant ataskaitas galima rinktis įvairias dokumento išdėstymo formas, šablonus, grupavimo ir duomenų rikiavimo būdus. Yra galimybė nustatyti lapo dydį, jo padėtį ir paraštes.

Pasitelkiant duomenų bazėje pateiktus lentelių ir užklausų duomenis galima sudaryti ir spausdinti įvairiausių ataskaitų projektus.

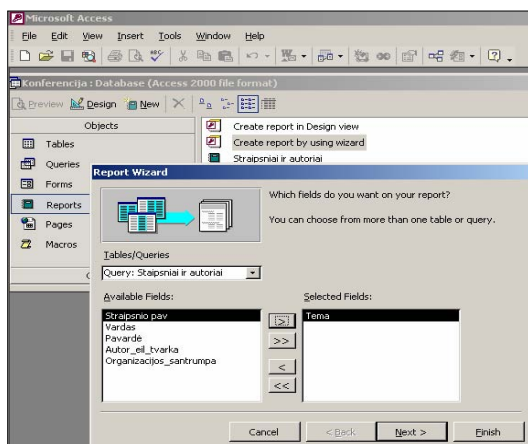
Kaip išspausdinti sudarytą ataskaitą pagal užklausos duomenis?

Mūsų tikslas – kokybiškai išspausdinti konferencijoje dalyvaujančių autorių sąrašą pagal temas ir skaitomus pranešimus bei autorių eilės tvarką.


Duomenų bazėje šie duomenys jau yra. Jie suformuoti praktiniame darbe, nurodančiame, kaip vykdyti kelių lentelių užklausą *Straipsniai ir autoriai*.

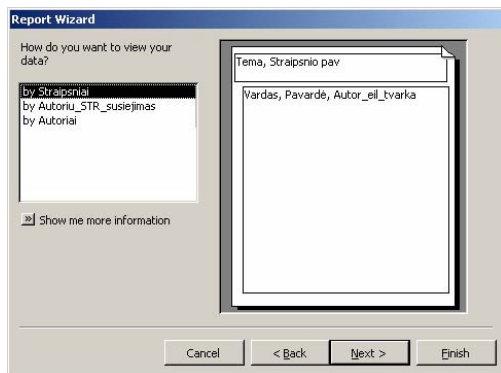
Esant aktyviam duomenų bazės objektui *Reports* dukart spaudžiant kairiąją pelės klavišą pasirenkamas ataskaitos formavimo būdas *Create report by using wizard*.

Pasirinkus atskaitos kūrimo priemonę – ataskaitų vedlį – ir atsisėrus langui *Report Wizard*, šiame lange pasirenkamas duomenų šaltinis, pavyzdžiui, jau anksčiau sukurta užklausa *Straipsniai ir autoriai*.



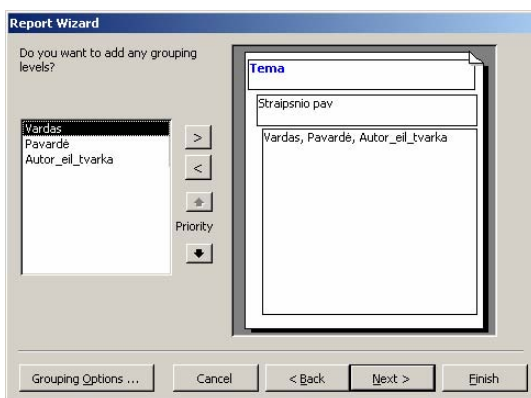
3.77 pav. Formuojamos ataskaitos laukų parinkimas

Šiame lange pažymimi visi ataskaitai sudaryti reikalingi laukai. Užklauso *Straipsniai ir autoriai* laukai pažymimi spaudžiant  mygtuką, jie perkeliami į *Selected Fields* sritį, tai atlikus spaudžiamas mygtukas *Next* ir atliekamas kitas žingsnis – parenkami grupavimo požymiai (3.78 pav.).




3.78 pav. Grupavimo požymių pateikimas

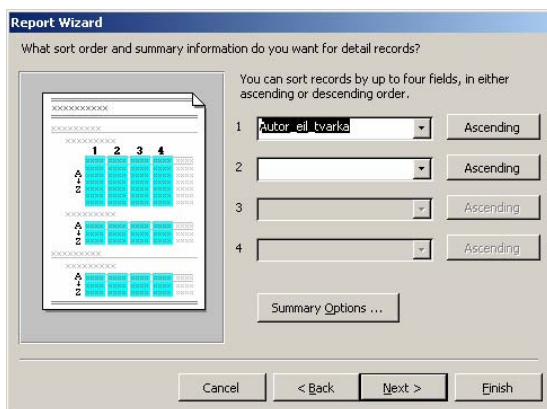
Spaudžiant mygtuką *Next* pereinama į grupių lygių parinkimo langą.



3.79 pav. Grupių lygių pateikimas

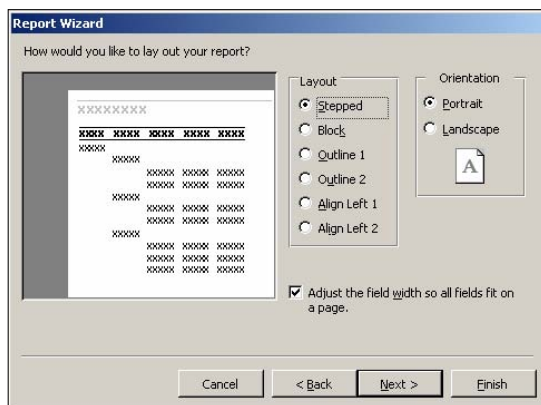
Atliekant šį žingsnį parenkami grupavimo požymiai (3.79 pav.).

Pavyzdžiui, lauką *Tema* išskiriame kaip aukščiausio lygio grupę, pažymėdami jį ir spausdami  mygtuką. Tai atlikus spaudžiant mygtuką *Next* pereinama į kitą žingsnį.



3.80 pav. Duomenų rikiavimo tvarkos ataskaitoje parinkimo langas

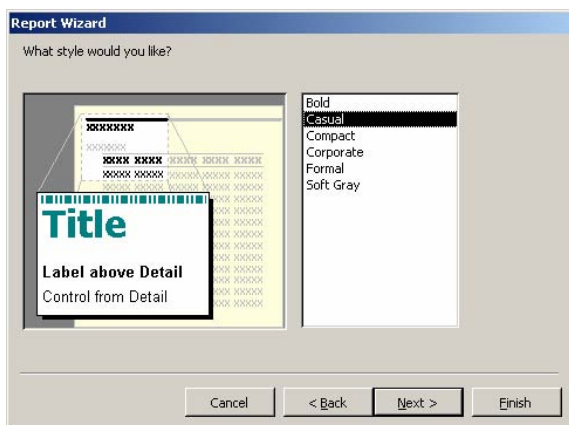
Šis *Report Wizard* langas (3.80 pav.) skirtas parinkti duomenų rikiavimo tvarką. Jo kairėje matyti ataskaitos vaizdas, dešinėje – keturi laukai, skirti rikiavimo požymiams nustatyti. Mygtuku *Ascending* nurodoma duomenis išdėstyti abėcėlės tvarka. Paspaudus šį mygtuką jo užrašas pasikeičia į *Descending*, tuomet parinktieji duomenys išdėstomi priešinga negu abėcėlės tvarka.



3.81 pav. Ataskaitos išdėstymo lape būdai

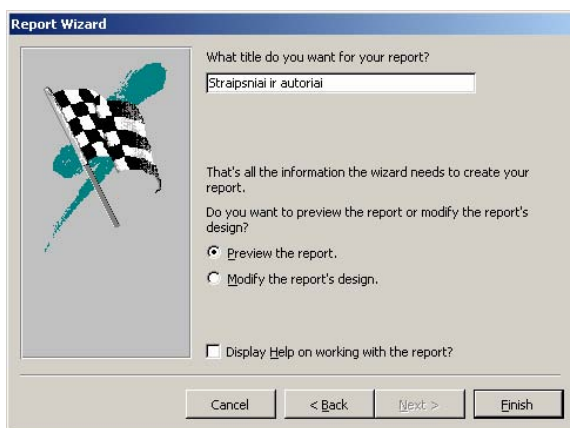
Ataskaitų apipavidalinimas

Apipavidalinimo (*Layout*) srityje galima parinkti vieną iš šešių ataskaitos išdėstymo lape būdų. Pasirenkamas būdas yra pateikiamas kairėje lango dalyje. Dalyje *Orientation* pasirenkama vertikali (*Portrait*) arba horizontali (*Landscape*) lapo padėtis. Įjungtas jungiklis *Adjust the field width so all fields fit on a page* rodo, kad ataskaitos matmenys turi tilpti popieriaus lape.



3.82 pav. Duomenų apipavidalinimo ataskaitoje parinkimo langas

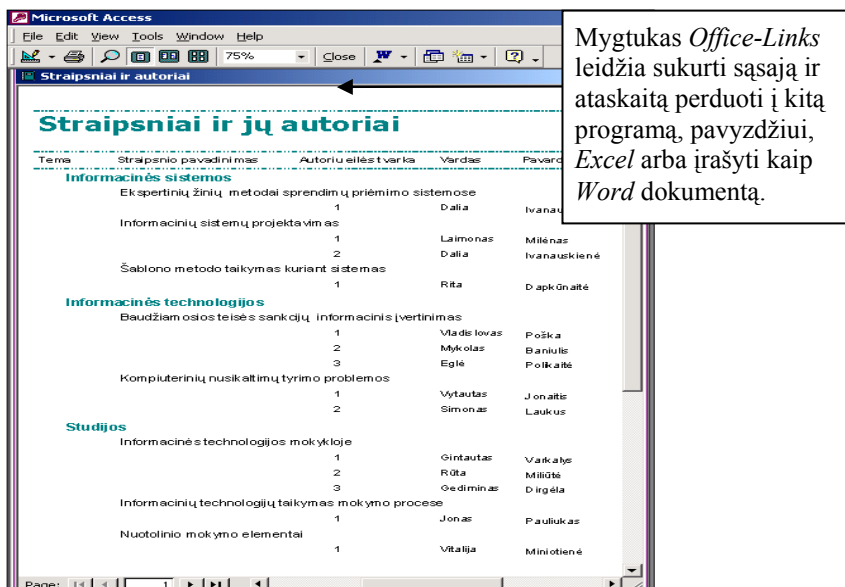
Šiame *Report Wizard* lange parenkami antraščių ir teksto šriftai, kairėje pusėje pateikiamas pasirinktojo šrifto vaizdas.



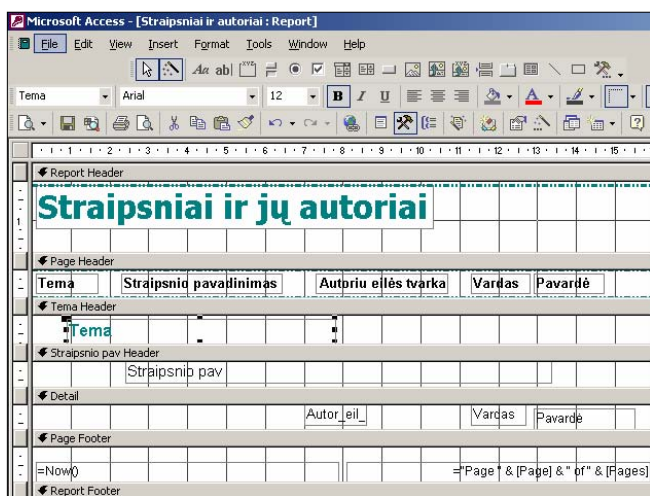
3.83 pav. Vedlio langas ataskaitai peržiūrėti ir formuluoti

Paskutiniame vedlio lange sudarytai ataskaitai suteikiamas pavadinimas ir paspaudžiamas mygtukas *Finish*.

Esant aktyviam *Preview the report* parametru (3.83 pav.), paspaudus *Finish* ataskaitą bus galima peržiūrėti.



3.84 pav. Ataskaitos peržiūros langas

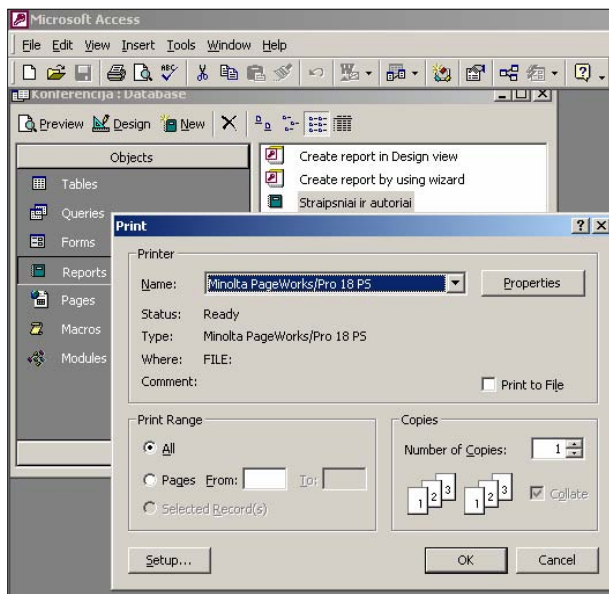


3.85 pav. Ataskaitos projekto langas

Jeigu netenkina duomenų padėtis lape arba kurie nors ataskaitos duomenys yra nereikalingi, esant suformuotai ataskaitai *Straipsniai ir autoriai*, paspaudus priemonių juostos mygtuką *Design* galima iškviesti projekto langą ir atlikti ataskaitos redagavimo veiksmus.

Etiketės tekstus galima redaguoti tiesiogiai įrašant simbolius ar pakeičiant juos naujais. Kad ataskaitos projekto lange esantys laukai vienas kito neuždengtų, galima sumažinti šrifto dydį arba padidinti įvedamo lauko plotį.

Ataskaitoje galima naudoti bazės funkcijas. Jas galima atverti priemonių juostos mygtuku *Build*. Ataskaitą galima išspausdinti vykdant komandą *File*→*Print*. Dialogo lango *Print* lauke *Name* parenkamas spausdintuvas, kuriuo bus spausdinama ataskaita.



3.86 pav. Spausdinimo parametrų nustatymas

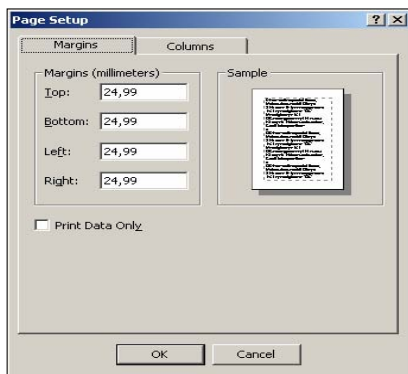
Dalyje *Print range* parenkama spausdinamos ataskaitos dalis, nurodanti, kas bus spausdinama:

- ✚ Visi ataskaita – *All*;
- ✚ Ataskaitos lapai, kurių lapų numerių intervalas yra nurodomas laukuose *From* (*nuo*) *To* (*iki*).

Dalyje *Kopijų skaičius* (*Number of Copies*) nurodomas ataskaitos kopijų skaičius.

Jungikliu *Collate* nurodoma ataskaitos spausdinimo tvarka, jeigu spausdinama ne viena kopija.

Mygtuku *Setup* atveriamas paraščių parinkimo langas.

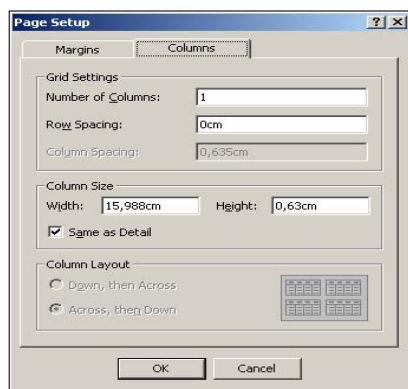


3.87 pav. Paraščių matmenų parinkimas

Nustatomos lapo paraštės: viršaus – *Top*; apačios – *Bottom*; kairioji – *Left*; dešinioji – *Right*.

Jungikliu *Print Data Only* įjungiamas juodraštinis spausdinimo variantas. Etikečių duomenys, linijos ir kiti dekoratyviniai elementai nespausdinami.

Skyrelyje *Stulpeliai* (*Columns*) galima pasirinkti ataskaitos duomenų išdėstymo būdą.



3.88 pav. Ataskaitos duomenų išdėstymas stulpeliais

Kontroliniai klausimai

1. Kam skirtos duomenų bazių ataskaitos?
2. Kokia ataskaitų kūrimo seka?
3. Ar galima išsaugoti sudarytą ataskaitą?
4. Kaip vykdoma ataskaitos peržiūra?
5. Kaip ataskaitą išspausdinti popieriuje?
6. Kaip iškviešti ataskaitos projekto langą ir kokius jau sudarytos ataskaitos pakeitimus galima atlikti projekto lange?
7. Kam skirtas reiškinis konstruktorius (*Build*)?
8. Kaip reiškinį konstruktoriumi parinkti parametrų lyginimo ir kitas duomenų apdorojimo funkcijas?

3.8. INFORMACIJOS PATEIKIMO PRAKTINIAI DARBAI

3.8.1. Serijinių laiškų rašymas naudojant duomenų bazę

1. Darbo tikslas

Šio praktinio darbo tikslas – išmokti automatiškai rašyti serijinius laiškus pasitelkiant duomenų bazės adresus.

Serijiniams laiškam būdinga tai, kad pagrindinis tekstas nekinta ir yra kartojamas visuose laiškuose, tik įrašant skirtingus rekvizitus, pavyzdžiui, adresą, gavėjo vardą, pavardę, datą.

Norint automatiškai įrašyti reikiamus rekvizitus, galima pasinaudoti sukurtos duomenų bazės, skaičiuoklių bei teksto redaktorių ir lentelių duomenimis.

2. Darbo užduotys





1. Iškvieskite *MS Word* programą.
2. Paruoškite laiško spausdinimo šabloną. Atitinkamai parinkite spausdinamo lapo dydį, paraštes, autoriaus rekvizitų išdėstymo vietą bei laiško tekstą.

3. Surinkite laiško šablono tekstą (teksto pavyzdys pateiktas praktinio darbo pabaigoje) ir išsaugokite jį, suteikdami failui pavadinimą, pavyzdžiui, *sablonasAM.doc*.
4. Iš duomenų bazės įterpkite užpildomų rekvizitų laukų pavadinimus, pasinaudokite konferencijos duomenų bazėje esančia lentele *Autoriai*. Failą išsaugokite diske (D:) aplanke *Studentas* arba diskelyje (A:).
5. Susiekite laiško šablona ir duomenų bazės lentelę *Autoriai* ir parenkite laiškų rinkinį, skirtą visiems autoriams, esantiems duomenų bazės lentelėje *Autoriai*. Parengtąjį laiškų rinkinį išsaugokite suteikdami failui prasminį pavadinimą, pavyzdžiui, *laiskai_visiems.doc*.
6. Susiekite laiško šablona ir duomenų bazę parengdami laiškų rinkinį, skirtą keliems adresatams, esantiems duomenų bazėje, pavyzdžiui, kurių įrašų numeriai nuo 4-to iki 6-to imtinai. Išsaugokite konkretaus laiškų rinkinio failą, suteikdami prasminį pavadinimą, pavyzdžiui, *laiskai_4_6.doc*.
7. Prieš spausdindami peržiūrėkite, kaip atrodo laišakai.
8. Išsiaiškinkite, kaip sukurtą laišką galima būtų išsiųsti elektroniniu paštu, pavyzdžiui, pirmajam autoriui. Jūsų kompiuteryje turėtų būti instaliuota elektroninio pašto programa, pavyzdžiui, *Outlook Express*.
9. Baikite dirbti su *MS Word* programa.

3. Metodiniai nurodymai

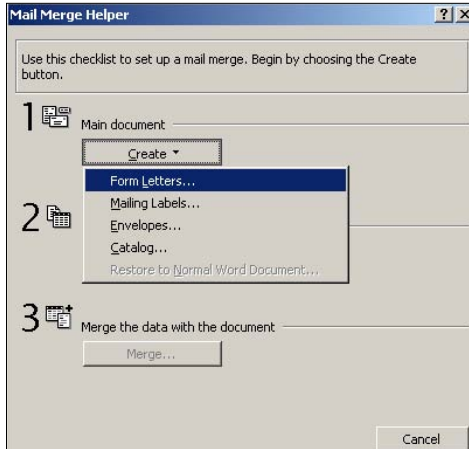
Kaip parinkti popieriaus lapo dydį, padėtį ir paraštes?

Įvykdykite komandą *File*→*Page Setup*. Iš dialogo lango pasirinkite skyrelį *Paper Size*. Iš išplėstinio sąrašo pasirinkite A4 formato lapo matmenis (210x297 mm). Puslapio padėties nustatymo (*Orientation*) grafoje pasirinkite *Portrait*, t.y. vertikalią lapo padėtį. Iš išplėstinio sąrašo *Apply To* pasirinkite *Whole Document* – visą dokumentą. Iš dialogo lango pasirinkite skyrelį *Margins*. Teksto įvedimo laukuose įrašykite, kokių matmenų paraščių norėsite:

-  *Top* – viršutinė paraštė – 3 cm,
-  *Bottom* – apatinė paraštė – 1 cm,
-  *Left* – kairioji paraštė – 2,5 cm,
-  *Right* – dešinioji paraštė – 1 cm.

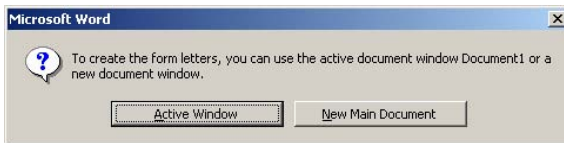
Kaip į laiško šabloną įterpti duomenų bazės laukų pavadinimus?

Į pasirinktą puslapį įrašykite pagrindinį laiško tekstą, kuris bus pastovus. Norint į laiško šabloną įterpti laukus, reikia vykdyti komandą *Tools*→*Mail Merge* ir atsivėrusiame lange *Mail Merge Helper* pasirinkti komandą *Form Letters*, skirtą kurti laiško formą.



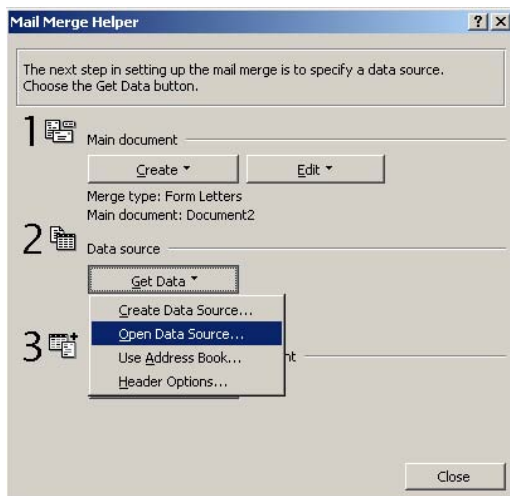
3.89 pav. Laiško formos pasirinkimo komandos *Tools*→*Mail Merge* dialogo langas

Jeigu aktyviame lange yra parašytas laiško šablonas, rinkitės *Active Window* režimą, priešingu atveju sukursite naują dokumentą (*New Main Document*).



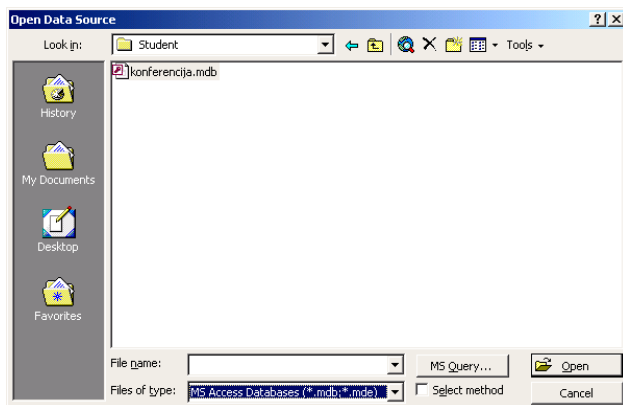
3.90 pav. Klausiamasis dialogo langas laiško šablonui kurti tame pačiame aktyvaus lango dokumente arba naujame dokumente

Vykdydami komandą *Open Data Source* atverkite duomenų šaltinį, kad galėtumėte iš anksčiau parengtų duomenų bazės laukų paimti adresato rekvizitų.



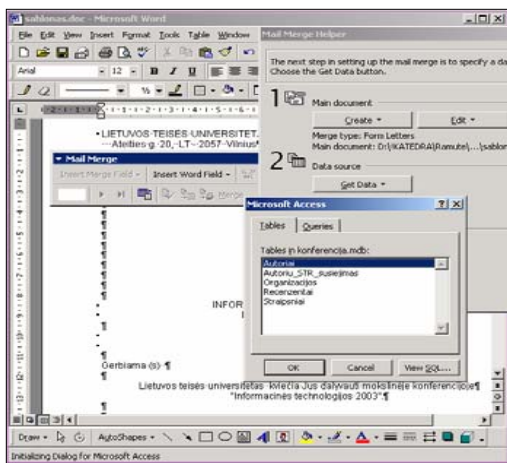
3.91 pav. Duomenų šaltinio parinkimas
(*Open Data Source*)

Open Date Source dialogo lange suraskite savo konferencijos duomenų bazės failą, pavyzdžiui, *konferencija.mdb*, ir jį atverkite.



3.92 pav. Duomenų šaltinio, papildančio sukurtą
laiško šablona, atvėrimo langas

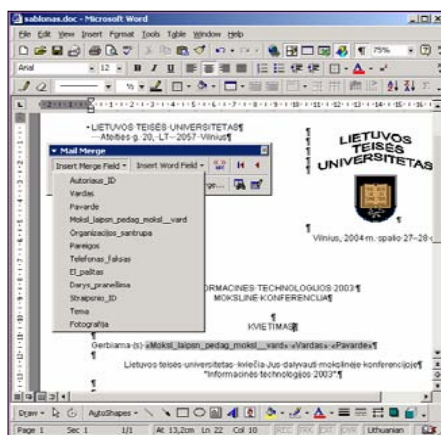
Atvertosios duomenų bazės struktūros *Microsoft Access* dialogo lange suraskite reikiamą lentelę (pvz., *Autoriai*) ir spauskite *Ok*.



3.93 pav. Lentelės *Autoriai* pasirinkimo ir jos laukų įterpimo į laiško šabloną pavyzdys

Programos *MS Word* langas pasipildys *Mail Merge* įrankių (mygtukų) juosta su pasirinktos duomenų bazės lentelės struktūra. Nustatykite teksto įvedimo žymiklį į atitinkamas rekvizitų vietas laiško šablone ir iš *Insert Merge Field* sąrašo pasirinkite tinkamus laukus (pvz., *Moksl_laipsn_pedagog_moksl_vard*, *Vardas*, *Pavardė*).

Toliau pateiktame šablono lange tamsesniame fone matyti jau įterpti laukų pavadinimai.



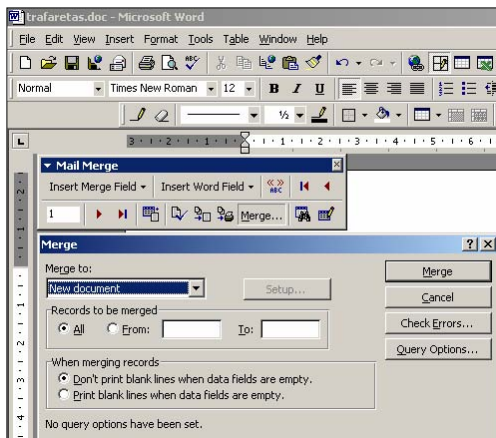
3.94 pav. Įterptamų lentelės *Autoriai* laukų sąrašas

Kaip susieti šabloną su duomenų bazės įrašais?

Jūsų sukurtas šablono failas ir failas, susietas su duomenų bazės įrašais, t. y. parengti laiškai, yra du skirtingi failai, turintys skirtingus pavadinimus.

Organizuodami darbą taip, kad šablonas ir parengti laiškų rinkiniai būtų saugomi skirtinguose failuose, galėsite prireikus panaudoti šabloną, jį papildyti ar keisti – parengti laiškus tik konkrečioms adresatams, juos redaguoti ir pan.

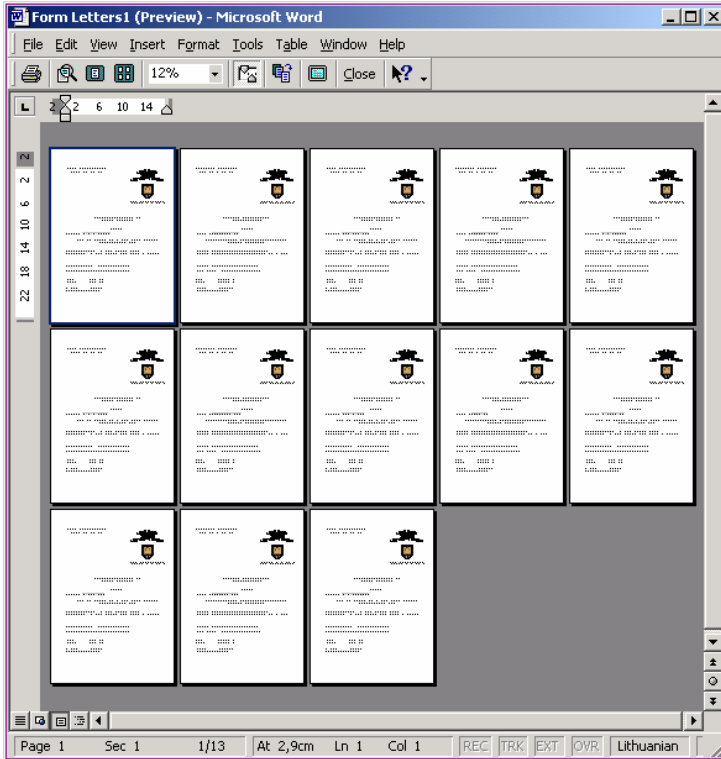
Paspauskite įrankių juostos *Mail Merge* mygtuką (*Merge*). Gautame komandos dialogo lange *Merge* pažymėkite *New document*. Rengdami laiškus visiems adresatams suaktyvinkite *All*, jeigu norite parengti laiškus konkrečiai adresatų grupei, į laukus įveskite duomenų bazės įrašų intervalą: *From (nuo) To (iki)*. Kai pasirinksite parametrus, vykdykite laiškų formavimo komandą *Merge*.



3.95 pav. Laiškų rengimo visiems autoriams dialogo langas

Kaip peržiūrėti laiškų rinkinio failą?

Jeigu darbą norite peržiūrėti, įvykdykite *File*→*Print Preview* komandą. *Microsoft Word (Preview)* lange paspauskite mygtuką *Multiple Pages* ir pasirinkite tiek puslapių, kiek laiškų esate parengę.

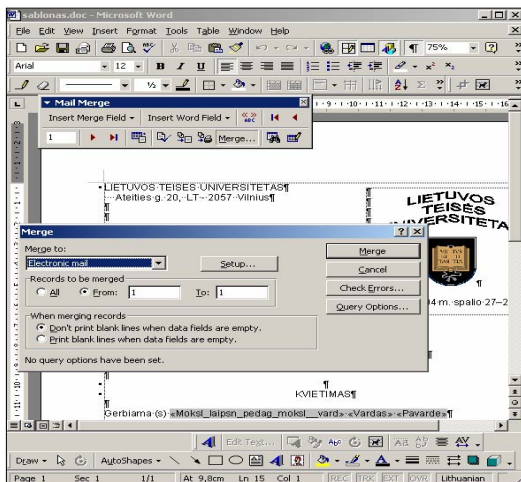


3.96 pav. Laiškų peržiūros langas

Kaip išsiųsti laišką elektroniniu paštu?

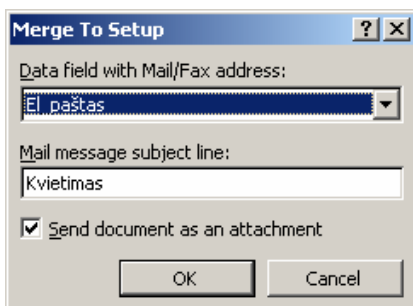
Norėdami susieti šabloną su duomenų bazės įrašais paspauskite mygtuką *Merge* (įrankių juostoje *Mail Merge*), lauko *Merge* to sąrašė pasirinkę *Electronic mail*, paspauskite *Setup* ir iš autorių lentelės pasirinkite elektroninio pašto adresą lauką. Kvietimą išsiųstus elektroniniu paštu kaip žinutės priedą.

Pavyzdyje matyti, kaip laiškas išsiunčiamas pirmam autoriui. Norint, kad laiškai būtų išsiųsti visiems autoriams, komandos dialogo lange *Merge* turi būti aktyvus *All* parametras.



3.97 pav. Laiško išsiuntimo elektroniniu paštu pavyzdys

Nustatomas duomenų bazės lentelės laukas, iš kurio bus imamos elektroninio pašto adresų reikšmės (3.98 pav.). Įvedama laiško tema.



3.98 pav. Elektroninio pašto lauko ir elektroninio laiško temos parinkimo langas

Spaudžiant mygtuką *Ok*, paskui – *Merge*, pasirodo programos *Outlook Express* langas.



3.99 pav. *Outlook Express* programos langas, kuriame mygtuku *Send* laiškas išsiunčiamas nurodytam adresatui

Programos *Outlook Express* lange spausdami mygtuką *Send* išsiųsite laišką nurodytu elektroniniu adresu.

Kaip baigti dirbti su programa MS Word?

Baigti dirbti su *MS Word* galima keliais būdais. Tam reikia:

- ✚ Įvykdyti lango valdymo meniu komandą *Close*;
- ✚ Įvykdyti komandą *File*→*Exit*.

Laiško šablono pavyzdys

LIETUVOS TEISĖS UNIVERSITETAS
Ateities g. 20, LT–2057 Vilnius



Vilnius, 2004 m. spalio 27–28 d.

INFORMACINĖS TECHNOLOGIJOS 2004
MOKSLINĖ KONFERENCIJA

KVIETIMAS

Gerbiama (s) «Moksl_laipsn_pedag_moksl_vard»
«Vardas» «Pavardė»

Lietuvos teisės universitetas kviečia Jus dalyvauti mokslinėje konferencijoje „Informacinės technologijos 2004“.

Konferenciją organizuoja Lietuvos teisės universiteto Valstybinio valdymo fakulteto Teisinės informatikos katedra.

Konferencijos vieta – Lietuvos teisės universiteto Valstybinio valdymo fakulteto Teisinės informatikos katedra, Ateities g. 20, 404 auditorija

Konferencijos pirmininkas

Mokslo sekretorė

Faksas (85)271 45 71

Telefonas (85)271 45 72

El. adresas infdep@ltu.lt

<http://www.ltu.lt/naujienos.html>

3.8.2. Interneto svetainės kūrimas programa *FrontPage*

1. Darbo tikslas

Šio praktinio darbo tikslas – išmokti projektuoti ir kurti interneto svetainę pasirenkant tematiką.

2. Darbo užduotys

1. Iškvieskite programą *MS FrontPage 2002*.
2. Pradėkite kurti interneto svetainę, į pasirinktą diską įrašydami jos aplanko pavadinimą.
3. Susipažinkite su pagrindiniu *FrontPage 2002* programos langu.
4. Paveikslėlius, kuriuos naudosite, perkelkite į *Images* aplanką tame pačiame diske.
5. Sudarykite dalyvio anketą ir ją perkelkite į kuriamos interneto svetainės aplanką.
6. Suplanuokite kuriamos interneto svetainės puslapių schemą, struktūrą, nustatykite hierarchijos lygmenis pagal pasirinktą tematiką, pavyzdžiui, informacijai apie konferenciją skelbti.
7. Apipavidalinkite pirmąjį svetainės puslapį, puslapio apačioje sukurkite nuorodas į žemesnio hierarchinio lygmens puslapius.
8. Pirmajame svetainės puslapyje sukurkite nuorodą į teksto redaktoriumi *MS Word* sukurtą dalyvio anketos failą.
9. Apipavidalinkite žemesnio lygmens puslapius, sudarydami juose lenteles, įrašykite pranešėjo vardą ir pavardę, pranešimo laiką ir pranešimo pavadinimą.
10. Išsiaiškinkite, kaip publikuoti sukurtą interneto svetainę, t. y. perkelti svetainę į tarnybinę stotį.

3. Metodiniai nurodymai

Kaip pradėti kurti interneto svetainę?

1. Įvykdykite naujos interneto svetainės kūrimo komandą *File→New→Page or Web*. Dialogo langelyje *New Page or Web* skyrelyje *New* pasirinkite naujo tinklalapio *Empty Web* parametą.

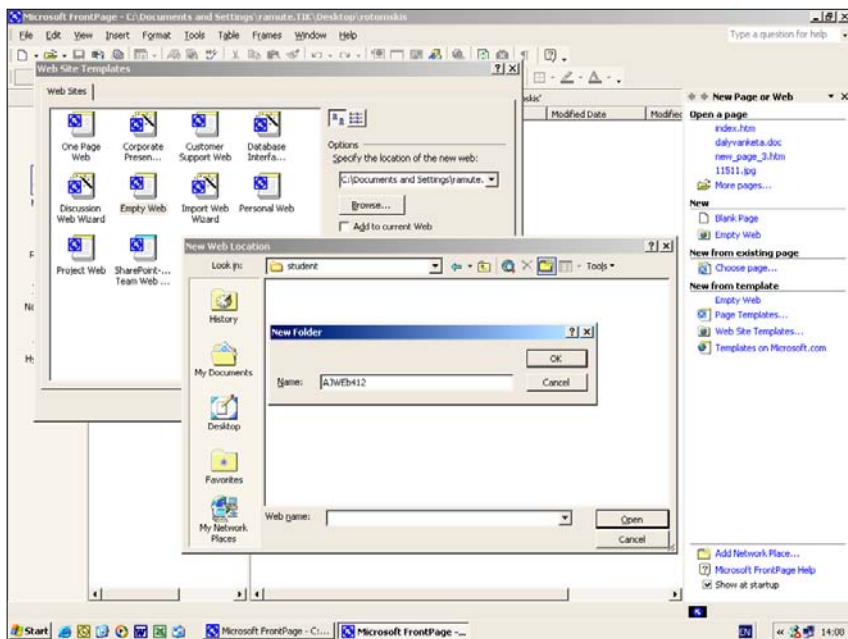
2. Dialogo lange *Web Site Templates* pasirinkite *Empty Web* parametą. Paspauskite peržiūros mygtuką *Browse*, jis atvers dialogo langelį *New Web Location*, jame įrašysite tinklalapio saugojimo vietos adresą. Dialogo langelyje *New Web Location* pasirinkite, pavyzdžiui, *D* diską, *Student* aplanką, paspauskite mygtuką *Create New Folder*.

3. Dialogo langelyje *New Folder* įrašykite aplanko pavadinimą, nurodydami savo grupę ir inicialus.

4. Paspauskite patvirtinimo komandas: *Ok*, *Open*, *Ok*.

Atlikus visus veiksmus, pasirinktajame diske bus sukurtas aplankas, kuriame bus saugomas kuriamas interneto svetainės puslapis, pavyzdžiui, *D:\Student\AJWEB412*.

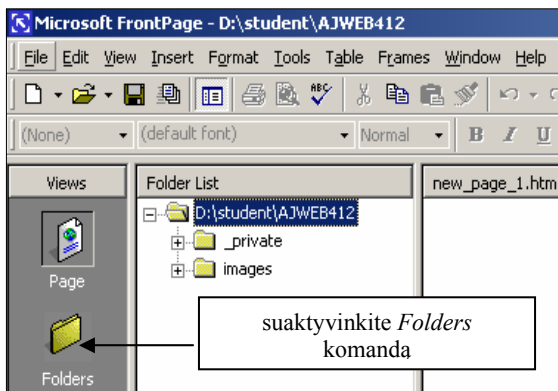
AJWEB412 – darbo aplanko pavadinimas, kuriame bus saugomi šiam puslapiui kurti reikalingi tekstiniai, grafiniai failai bei visa kita informacija, reikalinga kuriant interneto svetainę.



3.100 pav. Interneto svetainės darbo aplanko kūrimo langas

Suaktyvinkite aplanko *Folders* peržiūros komandą (3.101. pav.). Atsivėrus langeliui *Folder List* pamatysite, kad nurodytame

aplanke, pavyzdžiui, *AJWEB412*, programa sukūrė papildomus applanke. Aplankai, kurių pavadinimas prasideda brūkšneliu, reikalingi programos darbui, čia programa laiko jai reikalingą informaciją, jų keisti negalima. Į aplanką *Image* (atvaizdas, paveikslas) įrašykite visus paveikslėlius, kuriuos planuojate naudoti internetiniame puslapyje.



3.101 pav. Interneto svetainės darbo aplanko struktūra

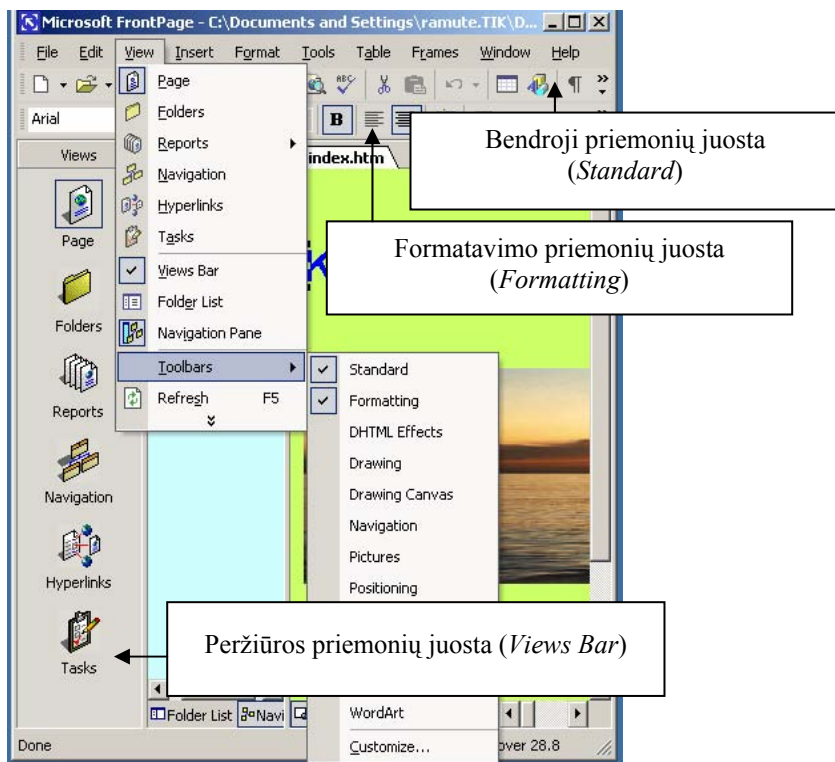
Susipažinkite su pagrindinėmis FrontPage 2002 programos priemonių juostomis.

Microsoft FrontPage lange pateikiamos pagrindinės įrankių juostos bei šių įrankių iškvietimo komanda *View*, esanti pagrindiniame meniu.

Mygtukai, esantys dešinėje peržiūros (*View Bar*) įrankių juostoje, leidžia greičiau pereiti iš vieno duomenų pateikimo būdo į kitą.

Peržiūros (*View Bar*) įrankių juostos mygtukai:

- *Page* – peržiūrėti ir keisti *Web* puslapį;
- *Folders* – peržiūrėti *Web* applanke ir failus, jų dydį, paskutinio keitimo datą;
- *Reports* – ataskaita apie bendrą failų, paveikslėlių ir nuorodų skaičių bei failų dydį;
- *Navigation* – interneto svetainės schemas peržiūra bei jos keitimo galimybė;
- *Hyperlinks* – interneto svetainės nuorodos;
- *Task* – užduočių sąrašas.



3.102 pav. Pagrindiniai *Front Page* programos lango elementai

Kaip sukurti konferencijos dalyvio anketą ir ją įkelti į interneto svetainės aplanką?

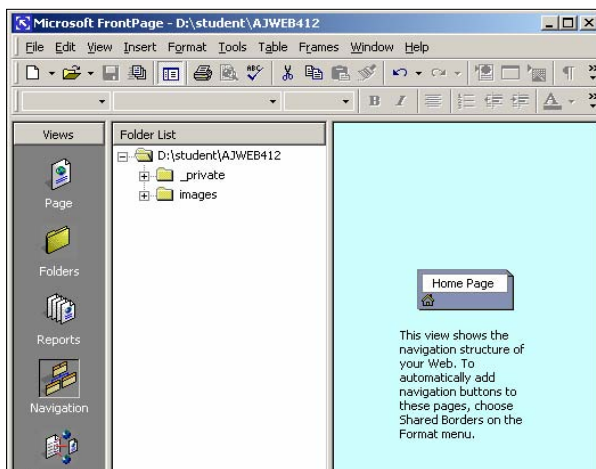
Į *AJWEB412* aplanką įkelkite dalyvio anketą, sukurtą programa *MS Word*. Jeigu jos nesate sukūrę, minimizuokite *FrontPage* langą, iškvieskite *MS Word* programą, sukurkite dalyvio anketą, dalyvio anketos failą nukopijuokite į *AJWEB412* aplanką.

Dalyvio anketos pavyzdys***Mokslinė konferencija „Informacinės technologijos“
Dalyvio anketa***

Vardas	
Pavardė	
Pedagoginis mokslo vardas ir mokslo laipsnis	
Organizacija	
Pareigos	
Adresas	
Telefonas, faksas	
Elektroninis paštas	
Pageidaučiau dalyvauti konferencijoje ir skaityti pranešimą šia tema	
Pranešimo bendraautoriai	
Vardas	
Pavardė	
Pedagoginis mokslo vardas ir mokslo laipsnis	
Organizacija	
Pareigos	
Adresas	
Telefonas, faksas	
Elektroninis paštas	
Pageidaučiau dalyvauti konferencijoje ir skaityti pranešimą šia tema	
Anotacija	
Straipsnyje nagrinėjami pagrindiniai informacinių sistemų kūrimo klausimai	

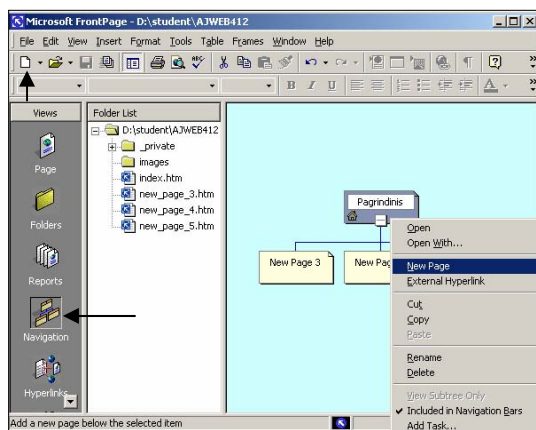
Interneto svetainės schemos sudarymas

Prieš pradėdami kurti interneto svetainę suplanuokite jos struktūrą. Mes siūlome kurti rengiamų konferencijų interneto svetainės puslapį. Tuo tikslu parengsime tokią svetainės schemą. Bus kuriamas pagrindinis svetainės puslapis bei žemesnio lygmens konferencijų tematiką atitinkantys puslapiai, kuriuose skelbsime informaciją apie rengiamas konferencijas.



3.103 pav. Interneto svetainės schemas kūrimas vykdant komandą *View*→*Navigation* ir pagrindinio puslapio įkėlimas

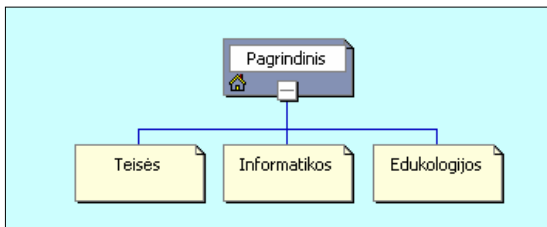
Prieš pradėdami kurti svetainės schemą iškvieskite puslapio išdėstymo projektą, paspauskite *Views Bar* įrankių juostos mygtuką *Navigation* arba vykdykite komandą *View*→*Navigation*. Pirmąjį – pagrindinį puslapį (*Home Page*) formuodami svetainės struktūrą įkelsime spausdami bendrosios (*Standard*) mygtukų juostos mygtuką *New*.



3.104 pav. Žemesnio lygmens puslapių įkėlimas į kuriama interneto svetainės schemą pasitelkiant kontekstinį meniu

Norėdami įkelti žemesnio lygmens puslapį, dešiniuoju pelės klavišu paspauskite aukštesnio lygmens, pavyzdžiui, pagrindinį (*Home Page*) puslapį ir vykdydami kontekstinio *Meniu* komandą *New Page* įkelkite žemesnio lygmens puslapius.

Puslapių hierarchijos lygmenį galima keisti juos perkeliant pele į atitinkamą vietą. Vizualiai bus matyti nauja hierarchija.

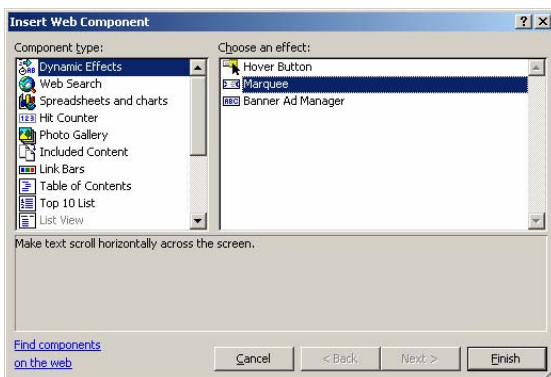


3.105 pav. Interneto svetainės schemas pavyzdys

Kaip kuriamas pagrindinis puslapis (Home Page) ir žemesnių lygmenų puslapių nuorodos?

Parenkamas puslapio fonas ir parašomas tekstas, kuris tiktų pagrindiniam puslapiui.

Pavyzdžiui, pagrindiniame puslapyje parašome tekstą KONFERENCIJOS KONFERENCIJOS KONFERENCIJOS. Norint, kad tekstas judėtų, reikia jį pažymėti ir vykdyti komandą *Insert* → *Navigation...* Komandos dialogo lange *Insert Web Component* parinkite judančio komponento rūšį *Dynamic Effects*, pasižymėkite slenkančios eilutės efektą *Marquee*, paspauskite *Finish*.

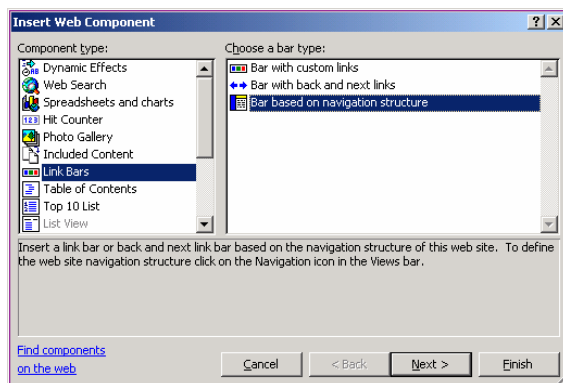


3.106 pav. *Insert* → *Navigation* komandos langas ir parinktas slenkančios eilutės efektas, skirtas pažymėtam tekstui

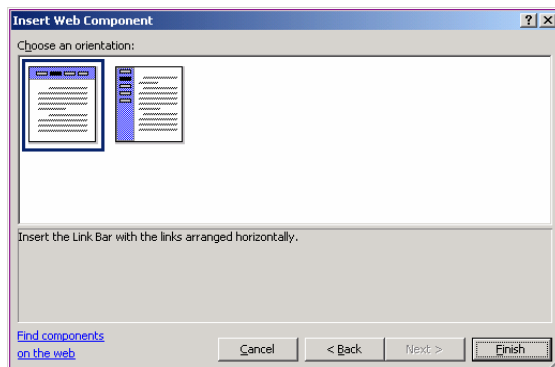
Puslapio fonas parenkamas vykdant komandą *Format*→*Background*. Norint įterpti paveikslėlius reikia vykdyti komandą *Insert*→*Picture*→*From File*.

Paveikslėliai turėtų būti įrašyti į *Images* aplanką. Jeigu neįrašėte paveikslėlių, tai galite padaryti ir dabar. Suradę paveikslėlio failą paspauskite dešinįjį pelės klavišą ir vykdykite komandas *Edit*→*Copy* ir *Edit*→*Paste Images* aplanke, neuždarydami programos *FrontPage* lango.

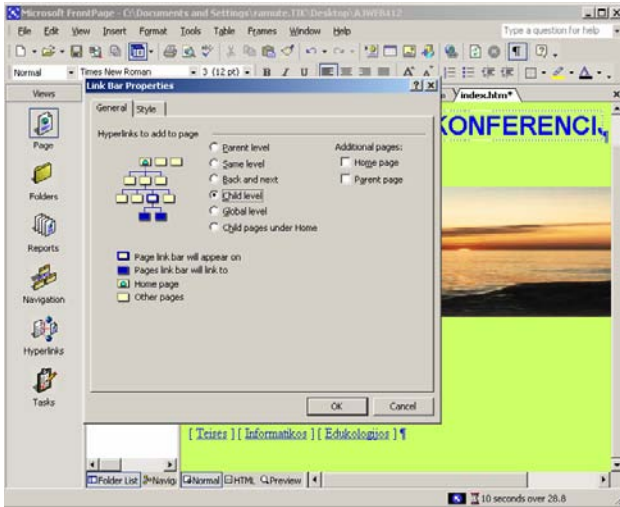
Norėdami pateikti nuorodas iš pagrindinio puslapio į žemesnio lygmens puslapius žymiklį nustatykite toje svetainės hierarchijos struktūros vietoje, kur turėtų būti šios nuorodos, ir vykdykite komandą *Insert*→*Navigation*...



3.107 pav. *Insert*→*Navigation* komandos langas, kuriame pažymėti parametrai nuorodoms kurti



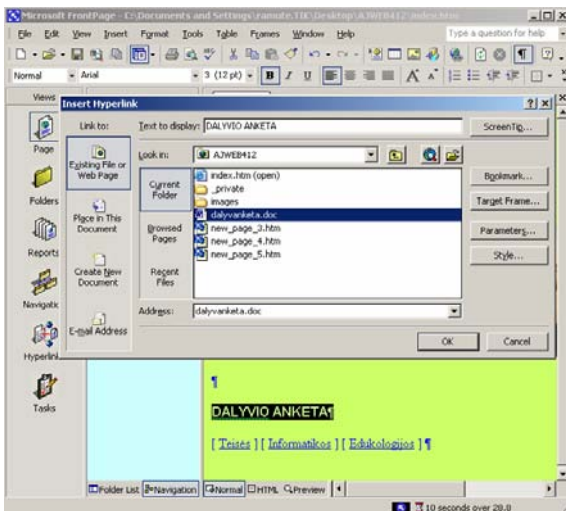
3.108 pav. Horizontalus ir vertikalus nuorodų pateikimo būdas



3.109 pav. Interneto svetainės nuorodų į žemesnio lygmens puslapius pasirinkimo pavyzdys

Pažymėkite žemesnio hierarchijos *Child level* lygmens puslapius ir vykdykite komandą.

Kaip sukurti nuorodą į tekstinį failą?

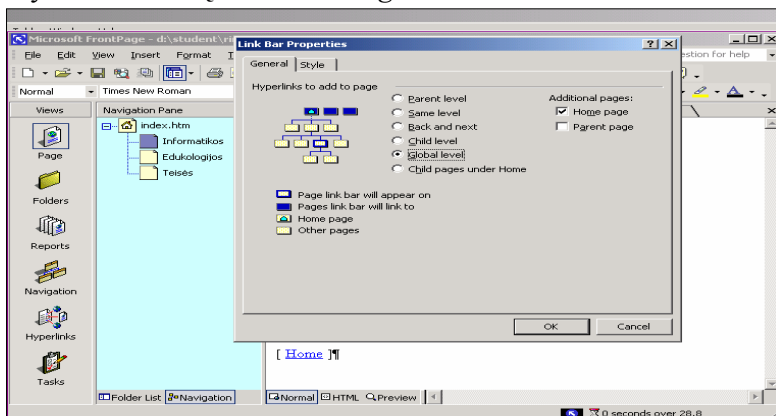


3.110 pav. Nuorodos į tekstinį failą *Dalyvio anketa* kūrimo pavyzdys

Parašykite tekstą, kuris pagal prasmę įvardytų jūsų nuorodą, pavyzdžiui, *Dalyvio anketa*. Šį tekstą pažymėkite ir vykdykite komandą *Insert*→*Hyperlink*, tada pažymėkite failą *Dalyvio anketa* ir vykdykite komandą spausdami *Ok*.

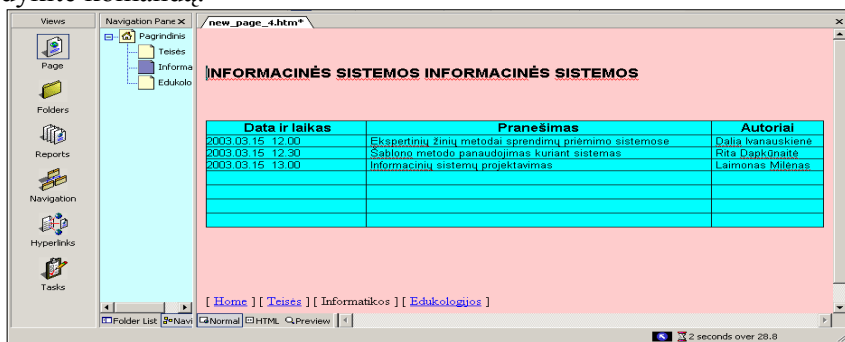
Kaip kuriama pagrindinio puslapio nuoroda žemesnio hierarchijos lygmens puslapiuose?

Norėdami pateikti nuorodas iš žemesnio lygmens puslapio į pagrindinį (*Home Page*) savo svetainės hierarchijos struktūros schemoje žymiklį nustatykite toje vietoje, kur turėtų būti ši nuoroda, ir vykdykite komandą *Insert*→*Navigation*.



3.111 pav. *Insert*→*Navigation* komandos dialogo langas, kai kuriama nuoroda iš žemesnio lygmens puslapio į pagrindinį

Pažymėkite aukščiausio lygmens pakopą *Global level* ir vykdykite komandą.

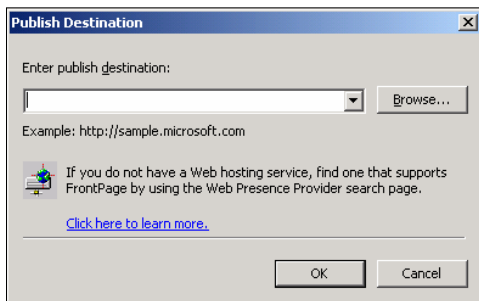


3.112 pav. Žemesnio lygmens puslapis interneto svetainėje, kuriame yra nuorodos į pagrindinį ir į to paties lygmens puslapius

Taip pat galima pasirinkti to paties lygmens variantą (*Same level*). Tuomet bus matyti nuorodos ir į to paties lygmens puslapius.

Kaip perkelti sukurtą interneto svetainę į tarnybinę stotį?

Norint svetainę publikuoti internete reikia ją perkelti į tarnybinę stotį (*Server*) vykdant komandą *File*→*Publish Web*.



3.113 pav. *File*→*Publish Web* komandos dialogo langas, perkiant svetainę į tarnybinę stotį

Veikiant protokolui *HTTP*, dialogo langelyje *Publish Destination* lauke *Enter publish destination* nurodykite *WWW* adresą (jis prasidės *http://*), kuriame įterpsite interneto svetainę. Nurodžius vartotojo kodą ir slaptažodį sukurta interneto svetainė bus perkelta į tarnybinę stotį (*Server*).

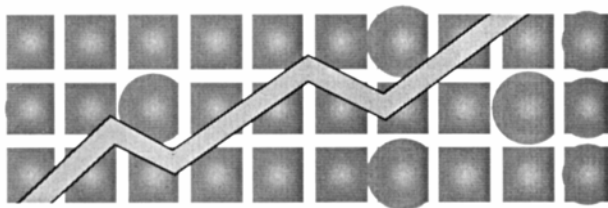
Jeigu tarnybinės stoties ryšys vykdomas *FTP* protokolu, dialogo langelyje *Publish Web* laukelyje *Specify The location* adresas prasidės *ftp://*.

Kontroliniai klausimai

1. Kaip valdomas programos *Microsoft FrontPage 2002* darbo langas ir jo sudėtinės dalys?
2. Kaip atliekamas interneto svetainės projektavimas, nustatant jos puslapių hierarchiją?
3. Nuorodų į interneto svetainės puslapius kūrimas atsižvelgiant į puslapių hierarchiją.
4. Kaip apipavidalinti svetainės puslapį naudojant foną bei judantį tekstą?
5. Kaip sukurti nuorodą į tekstinį failą?

4 skyrius

DUOMENŲ TVARKYMAS SKAIČIUOKLĖSE



- 4.1. Šiuolaikinės elektroninių lentelių tvarkymo galimybės
- 4.2. Skaičiuoklių darbo aplinka
- 4.3. Skaičiuoklių duomenų tipai ir formatai
- 4.4. Uždavinių sprendimo tvarka skaičiuoklėse
- 4.5. Darbo knygos sandara ir jos kūrimas skaičiuoklėse
- 4.6. Formuliu sudarymas skaičiuoklėse
- 4.7. Funkcijų panaudojimas
- 4.8. Duomenų bazių ir sąrašo analizės komandos
- 4.9. Klaidų reikšmės skaičiuoklių programoje
- 4.10. Grafinis duomenų pateikimas ir vizualizavimas
- 4.11. Skaičiuoklės darbo knygos spausdinimo būdai
- 4.12. Skaičiuoklių praktiniai darbai ir metodiniai nurodymai

4.1. ŠIUOLAIKINĖS ELEKTRONINIŲ LENTELIŲ TVARKYMO GALIMYBĖS

Daugelis vartotojų skaičiuokles (elektronines lenteles) naudoja paprasčiausioms lentelėms sudaryti, dažnai nenumanydami, kokios galimybės glūdi jų apdorojimui skirtoje programoje. Programa *Microsoft Excel* galima tvarkyti tekstą, valdyti duomenų bazes, atlikti sudėtingus skaičiavimus [1; 24; 26]. Šios programos galimybės yra tokios didelės, kad daugeliu atveju ji yra galingesnė negu specializuoti teksto redaktoriai ar duomenų bazių sistemos. Darbo pradžioje programos *Excel* funkcijų įvairovė gali nustebinti, tačiau įgijus praktinių įgūdžių ji bus neatskirama jūsų darbo dalis. Kaskart įgydami vis daugiau patirties suvokssite, kad skaičiuoklių aplinka skirta profesionaliam darbui, turi daug įvairiausių galimybių. Skaičiuoklių darbo aplinką galima priskirti prie koncepcinio modeliavimo programinės įrangos, turinčios funkcijų išraiškos galimybių, sprendėjo funkcijas, įgalinančias modeliuoti sprendimo priėmimo mechanizmą.

Programa *Excel* gali apdoroti labai dideles duomenų apimtis, atlikti įvairias operacijas su duomenimis – pradedant paprasčiausiu sumavimu ar vidurkio skaičiavimu ir baigiant sudėtingomis kompleksinėmis operacijomis, kurios užrašomos ne paprastais matematiniais simboliais, bet specialia programavimo kalba *Visual Basic* [26; 31].

Skaičiuoklės padeda dirbti buhalteriams, ekonomistams, statistikos specialistams. Jomis galima skaičiuoti lentelių struktūros duomenis, sudaryti diagramas, rinkti ir maketuoti tekstus. Informacija įrašoma ir išsimenama darbo knygoje (*Book*) – *Microsoft Excel* failuose, sudarytuose iš vieno ar kelių lapų (*Sheet*).

Naujose skaičiuoklės *Microsoft Excel* versijose elektroninės lentelės sąvoka keičiama kompiuterinės darbo knygos sąvoka. Kompiuterinę darbo knygą sudaro darbo lapai. Tai ne kelių lentelių rinkinys, o nauja priemonė, suteikianti galimybę taikyti erdvinius (trimačius) uždavinių sprendimo modelius ir metodus, atlikti sudėtingą matematinę ir statistinę duomenų analizę. Be visų turėtų dalykų, *Excel* yra labai patogi ir paprasta naudoti tiek pradedantiems, tiek

profesionaliems vartotojams. Sistema *Microsoft Excel* vartotojui suteikia išties plačias galimybes.

Duomenys įvedami, apdorojami ir apskaičiuojami lentelėse, kurių pagrindinis elementas yra ląstelė. Lentelės ląstelės (angl. – *cells*; literatūroje naudojami terminai – langelis, narvelis, celė), gali būti automatiškai užpildomos pasikartojančiais arba dėsningai besikeičiančiais duomenimis (t. y. taikomas duomenų sekų sudarymo mechanizmas).

Galima rinktis daugybę funkcijų (jų taikymas žymiai palengvina darbą), skirtų ekonomikos, matematikos, statistikos, duomenų bazių, informatikos, logikos uždaviniams spręsti ir kt.

Lentelės sudedamos į darbo lapus, o jie komponuojami į darbo knygas (*Workbooks*).

Daug galimybių suteikia diagramų žynys (diagramų patarėjas, t. y. konstruktorius-pagalbininkas *Chart Wizard*);

Sudarytoje lentelėje galima greitai atlikti regresinę ir kitą statistinę duomenų analizę. Tiesiogiai iš darbo lapo galima ruošti skaidres ir jas demonstruoti. Kelių lentelių duomenų analizei palengvinti galima pasinaudoti duomenų sujungimo (*Data Consolidating*) mechanizmu.

Be minėtųjų galimybių, yra keletas sudėtingesnių priemonių, palengvinančių darbą su dideliu duomenų sąrašu. Sąrašas – tai nauja duomenų bazės forma skaičiuoklėse, jis naudojamas sprendžiant sudėtingas duomenų apdorojimo užduotis. Yra duomenų tvarkymo komandų grupė, leidžianti atlikti sąrašo elementų rūšiavimo, filtravimo, tarpinių sumų sudarymo bei kai kurių duomenų slėpimo operacijas.

Microsoft Excel sistema turi puikias diagramų kūrimo galimybes. Jas taikyti gali tiek pradedantys vartotojai, tiek šios srities profesionalai. Skaičiuoklės diagramos gali būti arba dvimatės, arba trimatės. Diagrama – tai grafinis darbo lapo duomenų vaizdas. Pagal duomenų vaizdavimo formą diagramos skirstomos į tipus, pavyzdžiui, plotines diagramas, stulpelių, skritulines, žiedines, taškines, spindulines ir kitas. Kiekvienas diagramos tipas turi nuo 1 iki 4 modifikacijų ir nuo 4 iki 10 automatinių formatų. Automatinis vadinamas tam tikram diagramos tipui ir modifikacijai sukurtas formatas, turintis nustatytus keičiamus diagramos elementus.

Nepaprastai originalus *Microsoft Excel* pagalbininkas yra žy-

nys (*Wizard*). Taip pat yra sukurta naujų vartotojo sąveikos su programa priemonių. Funkcijų žynys aiškiai, profesionaliai pateikia funkcijų naudojimo ir formulių sudarymo galimybes. Pasitelkiant diagramų žynį tampa žymiai lengviau atlikti grafinę duomenų analizę. Jeigu sumanysite apskaičiuoti sudėtingesnio reiškinio rezultata, bakstelėkite pele žynio mygtuką ir ekrane pasirodžiusiame dialogo langelyje pasirinkite funkciją, kurios aprašymą ir argumentų paaiškinimus rasite kitame dialogo langelyje. Žynys paaiškins, kaip sudaryti funkciją, padės surinkti argumentus, apskaičiuos jų tarpines reikšmes, nurodys sintaksės, logikos formulių sudarymo klaidas.

4.2. SKAIČIUOKLIŲ DARBO APLINKA

Kaip ir bet kokia programa, veikianti langų valdymo aplinkoje, skaičiuoklės *Microsoft Excel* darbo aplinka valdoma pagal tuos pačius principus ir tokiomis pačiomis priemonėmis: meniu komandų grupės, įrankių juostos, dialogo langai, greitieji klavišai, atsakymus pažymintys mygtukai ir t. t. Skaičiuoklę vizualiai sudaro šios pagrindinės dalys: programos langas – stačiakampė ekrano sritis, kurioje atliekami veiksmai su vartotojo duomenimis ir dokumentais. Šis langas turi pavadinimo juostą (*Title Bar*), meniu juostą (*Menu Bar*), įrankių (priemonių) juostas (*Toolbars*), specialią formulių įvedimo eilutę (*Formula Bar*), dokumento sritį (*Document Area*) ir informacinę būsenos eilutę (*Status Bar*).

Dokumento srityje yra svarbiausias darbo elementas – dokumento langas, vadinamas darbo knyga (*Workbook*), joje yra vaizduojami vartotojo duomenys, formulės, diagramos ir visa kita, kas gali būti dokumente.

Darbo knygos duomenys kompiuterio atmintyje saugomi failuose (bylose), kurių plėtinys yra **.xls*, pavyzdžiui, *Apskaitos_darbo_knyga.xls*.

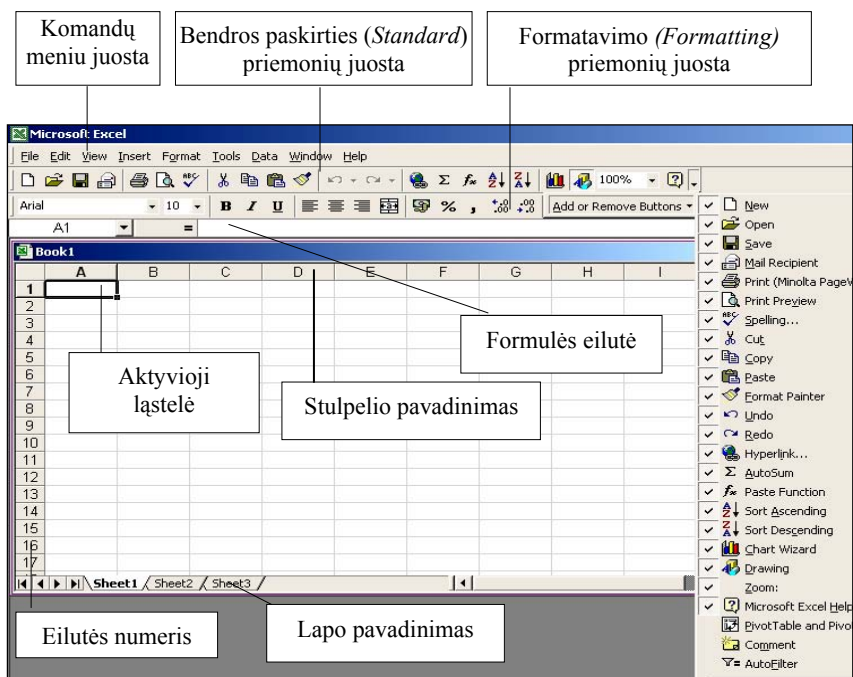
Susipažįstant su skaičiuoklės darbo aplinka reikėtų išsiaiškinti pagrindinius jos elementus.

Operacijos atliekamos darbo knygoje. Darbo knygą sudaro vienas arba keli darbo lapai (*Sheet*) – joje saugomi bei apdorojami duomenys. Darbo lapas – stačiakampė matrica, turinti 256 stulpelius (*Columns*) ir 65536 eilutes (*Rows*). Lapų skaičių leidžiama didinti.

Matricos elementai (eilučių ir stulpelių sankirtos) vadinami ląstelėmis (*Cells*). Kiekvieno stulpelio viršuje yra antraštė (*Column heading*), įvardijanti kiekvieną stulpelį lotyniška raide arba dviejų lotyniškų raidžių deriniu, pavyzdžiui: A, B, C, ..., AA, AB, ..., DZ, EA, ..., ZZ. Kiekvienos eilutės kairėje pusėje yra eilutės numeris arba antraštė (*Row heading*), įvardijanti eilutę. Eilutės numeris žymimas skaičiumi nuo 1 iki 65536. Netelpančiai į ekraną daliai pasiekti lango dešinėje pusėje ir apačioje yra dvi slankiosios juostos (*Scrollbars*).

Ląstelė (*Cell*) – lentelės bazinis elementas, esantis eilutės ir stulpelio sankirtoje.

Aktyvią lentelės ląstelę (*Active cell*) žymi stačiakampis tamsus rėmas, išskiriantis ją iš kitų. Aktyvioji ląstelė (*Active cell*) – tai darbo lapo ląstelė, su kurios turiniu yra šiuo momentu dirbama: į ją įvedami duomenys, formulė, ji redaguojama ir pan. Jeigu pažymėta daugiau kaip viena ląstelė – t. y. ląstelių grupė, vadinamasis ląstelių blokas (*Cell range*), aktyvi yra pirmoji pažymėta šio bloko ląstelė. Aktyvią ląstelę nurodo jos koordinatės.



4.1 pav. Pagrindiniai *Excel* programos lango elementai

Excel darbo knygoje gali būti ne tik darbo lapai su ląstelių matricomis, bet ir diagramų lapai (*Charts*), vartotojo dialogo langų lapai (*Dialog sheets*), *Visual Basic* kalba parašyti programų moduliai (*Modules*).

Lentelė (*Table*) – pirminis eilučių ir stulpelių struktūros dokumentas, kuriame informacija saugoma ląstelėse.

Sritis (*Range*) – lentelės ląstelių visuma, kurią sudaro stulpelio(-ų), eilutės(-čių) arba stačiakampio formos srities ląstelės.

4.2.1. Komandų ir kontekstinis meniu

Kaip ir kitos *Windows* aplinkoje veikiančios programos, *Excel* neapsieina be meniu juostos (*Menu bar*). Tai visiems kompiuterių vartotojams pažįstamas atveriamas (*drop-down*) meniu, kurį naudojant atliekamos pagrindinės *Excel* operacijos. Meniu juostoje yra devyni skyriai: *File*, *Edit*, *View*, *Insert*, *Format*, *Tools*, *Data* ir *Help*. Norint atverti meniu ir nurodyti reikiamą komandą, reikia pele bakstelėti vieno iš devynių nurodytų meniu pavadinimą ir pasirodžius komandų sąrašui pasirinkti norimą komandą. Meniu juostos ir komandų turinys gali keistis. Tai priklauso nuo to, kokio tipo lapas aktyvus. Šio lapo formą vartotojas gali keisti.

Meniu juostoje pasirenkamų komandų sąrašas skirtas pagrindinėms darbo funkcijoms atlikti:

- File* – darbo su lentelėmis ir jų failais komandos;
- Edit* – redagavimo komandos;
- View* – ekrano vaizdo komandos;
- Insert* – duomenų įterpimo komandos;
- Format* – formatavimo komandos;
- Tools* – darbo aplinkos konfigūravimo komandos;
- Data* – darbo su duomenų baze komandos;
- Window* – veiksmai su langais;
- Help* – aiškinamoji informacija (žinynas).

Be pagrindinio meniu, *MS Excel* yra kontekstinis meniu, iškviečiamas spaudžiant dešiniąją pelės klavišą. Koks kontekstinio meniu komandų sąrašas pateikiamas, priklauso nuo to, kurioje programos lango vietoje ir kokioje aplinkoje jis iškvieštas.

Meniu komandas vykdyti galima pele ir klaviatūra.

Norint failo atidarymo komandą vykdyti klaviatūra, reikia:

- ✚ Paspausti klavišą <Alt>;
- ✚ Klavišais <→> , <←> žymiklį nustatyti ties meniu *File*;
- ✚ Paspausti <Enter >;
- ✚ Klavišais <↑> , <↓> žymiklį nustatyti ties komanda *Open*;
- ✚ Paspausti <Enter >.

Šią komandą taip pat galima vykdyti spaudžiant klavišų derinį, t. y. <Ctrl> + <O>.

Norint pagreitinoti darbą reikia atsiminti dažniausiai naudojamų komandų klavišų derinius.

4.2.2. Ląstelių koordinacių tipai

Kiekvienos ląstelės padėtį *MS Excel* programoje rodo jos stulpelio ir eilutės koordinatės. Šios koordinatės – tai ląstelės adresas. Galima nurodyti vienos ląstelės, srities masyvo koordinates, pagal kurias surandami tose ląstelėse įrašyti duomenys. Ląstelių koordinatės gali būti santykinės, mišrios ir absoliučios.

4.1 lentelė. Ląstelių koordinacių tipai ir jų žymėjimas

Ląstelės koordinacių tipas	Žymėjimas	Ląstelių koordinacių taikymo formulėse paaiškinimas
Santykinės	A5	Ląstelių ir blokų santykinės koordinatės formulėje pakeičiamos per tiek stulpelių ir eilučių, per kiek stulpelių ir eilučių perkeliama nukopijuotoji formulė
Absoliučios	\$A\$5	Ląstelių ir blokų absoliučios koordinatės formulėje nekeičiamos
Mišrios	A\$5 arba \$A5	Kopijuojant formules atitinkamai keičiamos tik santykinės koordinacių dalys









Skirtingi ląstelių koordinacių tipai taikomi sudarant formules.

4.2.3. Įrankių juostos

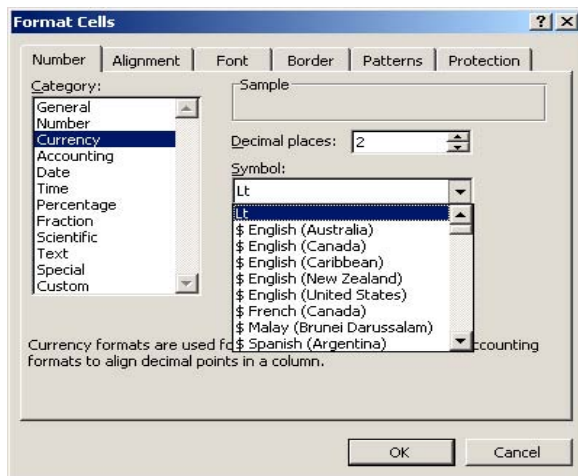
Įrankių juostos (*Toolbars*) gali būti vertikalios arba horizontalios judančios stačiakampių mygtukų juostos. Jos, kaip ir meniu komandos, skirtos valdyti *Excel*. Akylesnis vartotojas iškart supras, kad visi šie spalvoti mygtukai yra ne kas kita kaip grafiškai pavaizduotos meniu komandos. Pavyzdžiui, šrifto, stiliaus, šrifto dydžio, lygiavimo įrankiai tik paprastina vartotojo darbą: norint pakeisi ląstelės turinio šriftą nereikia kaskart jungti meniu *Format*→*Cells*→*Font*. Mygtukai, vykdantys komandas *New*, *Load*, *Save*, *Print*, *Print Preview*, *Cut*, *Copy*, *Paste*, *Undo*, *Redo*, yra ne kas kita kaip meniu darbo su failais (*File*) ir redagavimo (*Edit*) komandos. Mygtukai įrankių juostose grupuojami pagal paskirtį. Dažniausiai naudojamos *Standard*, *Formatting* ir *Drawing* įrankių juostos.

4.3. SKAIČIUOKLIŲ DUOMENŲ TIPAI IR FORMATAI

Kiekvieną lentelės ląstelę galima užpildyti duomenimis. Duomenys gali būti skirtingų tipų. Kokio tipo duomenys bus įrašyti į ląsteles, priklauso nuo pasirinkamų formatavimo galimybių. Ląstelėje esantys duomenys gali būti:

-  Skaičius;
-  Tekstas;
-  Valiuta;
-  Skaičiavimo pagal formules rezultatas;
-  Procentinė išraiška;
-  Data;
-  Laikas;
-  Pasikartojantis tekstas ir kt.

Ląstelėse esantys duomenys gali būti formatuojami vykdant komandą *Format*→*Cells*. Norint tinkamai apdoroti duomenis reikia teisingai pasirinkti duomenų formatą, jis gali būti bendras, skaičių, piniginis, datos, laiko, procentinis, trupmeninis, eksponentinis, teksto, specialusis, sukurtas vartotojo.



4.2 pav. Duomenų formato suteikimo komandos

Skaičiaus su dešimtaine trupmena formatu vaizduojami sveikieji ir realieji skaičiai. Esant šiam formatui gali būti nurodomas trupmeninių pozicijų skaičius bei tūkstančių skiriamasis simbolis, pavyzdžiui, tarpo ženklas arba taškas. Gali būti nurodomas ir neigiamo skaičiaus vaizdavimo būdas, skaičius gali būti apskliaudžiamas arba vaizduojamas kita spalva.

Pasirinkus procentų formatą, ląstelėje įrašytas skaičius dauginamas iš 100, o jo dešinėje įrašomas procentų ženklas. Nurodant šį formatą galima pasirinkti dešimtainių ženklų skaičių po kablelio.

Valiutos formatas naudojamas pinigų reikšmėms vaizduoti. Nurodomas pateiktojo skaičiaus valiutos simbolis.

Paprastųjų trupmenų formatai suteikia skaičiams paprastųjų trupmenų pavidalą. Esant šiam formatui nurodomas skaitmenų skaičius trupmenoje.

Eksponentinių skaičių formatas skirtas labai dideliems arba labai mažiems skaičiams vaizduoti. Šio formato žymuo $E+n$ reiškia skaičių, padaugintą iš 10^n .

Suteikus teksto formatą skaičiui, jis traktuojamas kaip tekstas ir pritraukiamas prie kairiojo ląstelės krašto.

Pasirenkant vartotojo formatą, galima sudaryti įvairių formatų šablonus, taip pat kurti individualius formatus. Norint sukurti indivi-

dualų formatą reikia:

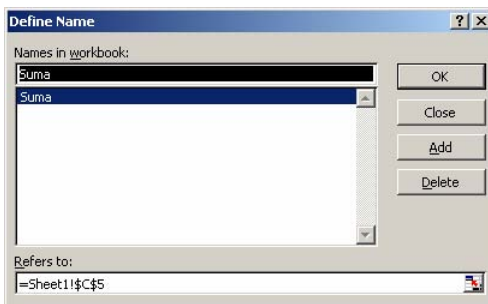
- ✚ Vykdyti komandą *Format*→*Cells*, pasirinkti skyrelį *Number*;
- ✚ Sąraše *Category* pažymėti *Custom* formatų grupę;
- ✚ Pateiktų formatų šablonų sąraše pažymėti tinkamiausią formatą;
- ✚ Redaguoti laukelyje *Type* pažymėtą formatą arba įvesti visai naują formato šabloną.

4.2 lentelė. Individualių formatų pavyzdžiai

Formatas	Įvestas skaičius	Rodomas skaičius
#.###,00	13000	13.000,00
0.000,00	13	0.013,00
#.	13000	13
“Sąsk.Nr.” 0000	13	Sąsk.Nr.00013
(00) 000 00 00	852714571	(85) 271 45 71
Lt* #.##0,00	1130	Lt 1.130,00

Ląstelei galima suteikti pavadinimą ją pažymint ir vykdant komandą *Insert*→*Name*→*Define*. Dialogo langelyje *Define Name* paspauskite mygtukus *Add* ir *OK*. Suteikus ląstelei pavadinimą ją galima naudoti formulėse vietoje ląstelės adreso.

Priskiriant ląstelei pavadinimą svarbu, kad jis prasidėtų raide ir nebūtų panašus į ląstelių koordinatas. Geriausia ląstelėms suteikti prasminius pavadinimus, tarkime, ląstelę, kurioje įrašytas dydis atitinka saugomos informacijos prasmę, pavyzdžiui, saugomą ketvirčio pinigų sumą, reikia pavadinti suma.



4.3 pav. Prasminio pavadinimo ląstelei suteikimas

Ląstelei suteikus prasminį pavadinimą, jis pakeičia ląstelės koordinatas.

	A	B	C	D
1				
2		sausis	800,00 Lt	
3		vasaris	990,00 Lt	
4		kovas	770,00 Lt	
5		Suma	2.560,00 Lt	
6				

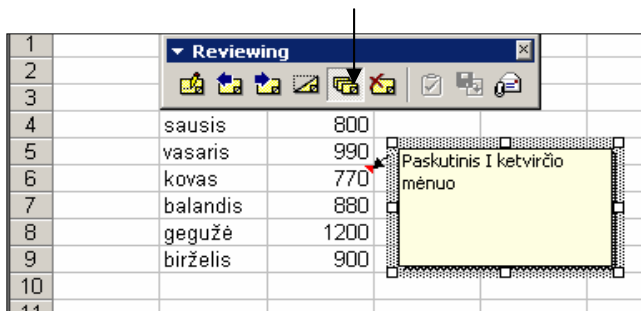
4.4 pav. Ląstelės koordinatų keitimas pavadinimu

Norint panaikinti suteiktą pavadinimą, reikia pažymėti ląstelę arba sritį, kurios pavadinimą reikia naikinti, ir vykdyti komandą *Insert*→*Name*→*Define*. Komandos dialogo langelyje reikia pažymėti ląstelės arba srities pavadinimą ir paspausti mygtukus *Delete* ir *Ok*.

Vykdyt komandą *Insert*→*Comment* ląstelėje gali būti rašomi komentarai. Atsiradus teksto rėmams įrašomas komentaras. Įrašius komentarą ląstelės dešinysis viršutinis kampas bus pažymėtas kita spalva.

Norint peržiūrėti visus komentarus reikia vykdyti komandą *View*→*Comments*, pasirodys įrankių juosta *Reviewing*.

Priemonių juostoje *Reviewing* yra aktyvus mygtukas *Show/Hide* parodantis arba paslepiantis visus komentarus.



4.5 pav. Duomenis paaiškinantis komentaras ir *Reviewing* priemonių juosta, skirta dirbti su komentarais

4.4. UŽDAVINIŲ SPRENDIMO TVARKA SKAIČIUOKLĖSE

Pirmiausiai atveriamas darbo lapas ir nurodomi pradiniai duomenys, tuomet įrašomos reikalingos formulės ir funkcijos. Jeigu lentelė bus naudojama daugiau nei vieną kartą ir joje bus įrašomi skirtingi duomenys, sukuriamas lentelės darbo lapo šablonas:

- ✚ Pasirenkamos lentelės antraštės, paraštės ir kt.;
- ✚ Įrašomi vienaarūšių duomenų stulpelių pavadinimai ir nubrėžiamos stulpelių linijos;
- ✚ Įrašomi pirminiai duomenys;
- ✚ Duomenims apskaičiuoti sudaromos formulės, o duomenis vizualiai pateikti – diagramos;
- ✚ Darbo lapas formatuojamas ir šablonas išsaugomas.

Pakartotinai sprendžiant uždavinį iškviečiamas jau sudarytas darbo lapo šablonas, jis redaguojamas įrašant naujus duomenis. Įvedant duomenis atliekama tarpinio duomenų išsaugojimo diske operacija. Prieš spausdinant dokumentą, jis peržiūrimas ekrane, parenkami spausdinimo parametrai.

4.5. DARBO KNYGOS SANDARA IR JOS KŪRIMAS SKAIČIUOKLĖSE

4.5.1. Skaičiuoklių darbo knygos kūrimas

Pradėjus darbą su *Excel* programa *Excel* darbo lange sukuriamą naują tuščią knygą, kuriai laikinai suteikiamas *Book1* pavadinimas. Dažniausiai pasirenkamas pavadinimas, kuris atspindės saugomos informacijos prasmę. Sukurti naują knygą galima ir darbo *Excel* programa metu, bendrosios paskirties (*Standard*) mygtukų juostoje spaudžiant naujos knygos mygtuką *New*.

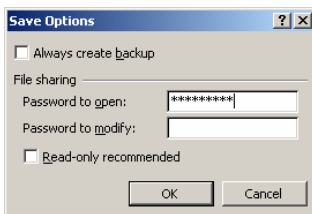
Naują dokumentą galima kurti ir vykdant *File*→*New* komandą. Taip galima pasirinkti dokumento šabloną, kuriuo naudojantis įvedami duomenys ir atliekami skaičiavimai.

Failo išsaugojimas diske

Jei į diską įrašoma nauja knyga arba senąją reikia išsaugoti nauju pavadinimu, vykdoma komanda *File→Save As*. Komandos dialogo lange reikia: išplėstiniame sąraše *Save in* pažymėti diską bei katalogą, kuriame reikia išsaugoti failą; teksto lauke *File name* įvesti bylos pavadinimą be plėtinio. Programa prie failo pavadinimo prideda plėtinius, pavyzdžiui, *.xls* – esant lentelėms, *.xlt* – šablonams.

Jeigu failą norima išsaugoti įslaptinant (nežinantiems slaptažodžio neleidžiama failo nuskaityti arba redaguoti), reikia:

- ✚ Vykdyti komandą *File→Save As*. Komandos dialogo lange *Save As* vykdyti *Tools→General Options*;
 - ✚ Lauke *Password to open* įrašyti pasirinktą slaptažodį, reikalingą failui nuskaityti;
 - ✚ Lauke *Password to modify* įrašyti pasirinktą slaptažodį, reikalingą failui redaguoti;
 - ✚ Slaptažodį pakartoti dialogo langelyje *Confirm Password*;
- Paspausti mygtuką *Save* ir dokumentas bus įrašytas į diską.



4.6 pav. Failo išsaugojimas, įrašant slaptažodį failui nuskaityti

Norint į diską įrašyti redaguojamą *Excel* dokumentą, kuris jau kartą buvo įrašytas, reikia paspausti bendrosios paskirties (*Standard*) mygtukų juostos mygtuką *Save* arba vykdyti komandą *File→Save*.

Failo įkėlimas

Norint įkelti iš disko bet kurį *Excel* failą, reikia:

- ✚ Vykdyti komandą *File→Open*;
- ✚ *Open* komandos dialogo lango išplėstiniame sąraše *Look in* nurodyti dokumento diską, katalogą ir išrinkti iš sąrašo failą bei jį atidaryti.

Failų, su kuriais buvo dirbta paskutiniu metu, sąrašas yra *File* meniu apačioje.

4.5.2. Ląstelės turinio pildymas ir redagavimas

Visi duomenys *Excel* programoje įrašomi į ląsteles. Norint rašyti į ląstelę, reikia:

- ✚ Pažymėti ląstelę, į kurią reikia įvesti duomenis;
- ✚ Įrašyti įvedamus duomenis: tekstą, skaitmenis, formules;
- ✚ Paspausti <Enter> arba pažymėti kitą ląstelę.

Jeigu ekrane matyti formulės eilutė, įvedami duomenys yra atkartojami šioje eilutėje, o eilutės kairėje atsiranda trys papildomi mygtukai. Tekstas rašomas tuo šriftu, kurio pavadinimas užrašytas šrifto lange, esančiame formatavimo mygtukų juostoje. Jeigu šis šriftas neturi reikalingų simbolių, pavyzdžiui, lietuviškų raidžių, prieš pradėdamas rašyti tekstą, jį reikia pakeisti kitu šriftu. Gali būti formatuojami ir jau įvesti duomenys.

Įvestus ląstelėse duomenis beveik visada tenka taisyti ir pertvarkyti. *Excel* dažniausiai atliekami tokie redagavimo veiksmai:

- ✚ Redaguojamas ląstelės turinys;
- ✚ Šalinami nereikalingi duomenys;
- ✚ Perkeliama duomenys;
- ✚ Kopijuojamos ląstelės.

Norint atlikti veiksmus su ląstelėje ar srityje esančiais duomenimis, juos reikia pažymėti.

4.3 lentelė. Ląstelių žymėjimo veiksmai

Žymimas objektas	Veiksmo aprašymas
Ląstelė	Spragtelėti ties išskiriama ląstele pele arba kryptinių rodyklių klavišais; į šią ląstelę perkelti lentelės žymiklį.
Sritis	Spragtelėti pele ties kampine srities ląstele ir traukti pelės rodyklę iki kitos srities kampinės ląstelės.
Eilutė arba stulpelis	Spragtelėti antraštėje ties eilutės numeriu arba stulpelio pavadinimu.
Sudėtinė sritis	Pažymėti pirmą sritį, tada laikant nuspaudus <Ctrl> klavišą, išskirti papildomas sritis.
Visa lentelė	Spragtelėti pele ties stačiakampiu, esančiu stulpelių ir eilučių pavadinimų sankirtoje (virš 1 eilutės į kairę nuo A stulpelio).
Lapas	Spragtelėti pele ties lapo pavadinimu lapų sąrašo eilutėje, esančioje programos lango apačioje.
Keli lapai	Laikant nuspaustą <Ctrl> klavišą, spragtelėti pele ties reikiamų lapų pavadinimais lapų sąrašo eilutėje.

Ląstelės turinio redagavimas

Ląstelės turinį redaguoti galima formulės eilutėje arba tiesiogiai ląstelėje.

Norint redaguoti duomenis formulės eilutėje reikia pažymėti redaguojamą ląstelę, spragtelėti pele reikiamoje formulės eilutės vietoje, redaguoti ląstelės turinį, baigus redaguoti paspausti klavišą <Enter> arba pereiti prie kitos ląstelės.

Norint redaguoti duomenis tiesiogiai ląstelėje reikia pažymėti redaguojamą ląstelę, paspausti funkcijų klavišą <F2> arba du kartus spragtelėti pele ląstelėje, redaguoti ląstelės turinį, baigus redaguoti – paspausti <Enter> arba pereiti prie kitos ląstelės.





Ląstelių turinio naikinimas

Norint panaikinti vienos ląstelės (arba srities) turinį, reikia pažymėti ląstelę (arba sritį), kurios turinį reikia panaikinti, ir paspausti <Delete> klavišą arba vykdyti komandą *Edit*→*Clear*→*All*.




Ląstelės (srities) turinio kopijavimas ir perkėlimas

Kopijuoti arba perkelti ląstelės turinį galima esant komandiniam režimui, naudojant laikinąją atmintį (*Clipboard*) ir vykdamas komandas *Copy*, *Cut* ir *Paste*.

Naudojant komandinį režimą, reikia:

-  Pažymėti kopijuojamą (arba perkeliama) ląstelę (arba sritį).
-  Vykdyti meniu *Edit* komandą *Copy* (*Cut*). Kopijuojama dalis apibrėžiama mirksinčia punktyrine linija.
-  Pažymėti ląstelę (arba sritį), į kurią reikia kopijuoti (perkelti) duomenis.
-  Vykdyti meniu *Edit* komandą *Paste*.

Norint kopijuoti arba perkelti ląstelės turinį pele, reikia:

-  Pažymėti perkeliama ląstelę (sritį);
-  Pelės žymiklį nustatyti ties ląstelės ribos linija, kad jis įgytų rodyklės pavidalą;
-  Laikant nuspaudus <Ctrl> ir kairįjį pelės klavišą (kopijuojant) arba spaudžiant tik kairįjį pelės klavišą (perkeliant), perkelti ląstelės turinį į kitą vietą.

4.5.3. Formulės įrašymas į ląstelę


Formulės sudaromos norint atlikti sudėtingesnius skaičiavimus. Sudaroma konkrečios ląstelės arba masyvo formulė. Vėliau ji užpildoma kitose ląstelėse, kurių rezultatas apskaičiuojamas pagal tą patį algoritmą. Formulėms sudaryti naudojamos operacijos, funkcijos, ląstelės dažniausiai žymimos jų koordinatėmis. Kiekviena formulė būtinai prasideda formulės redagavimo ženklu =. Ląstelėje gali matytis arba formulės reikšmė, tai yra apskaičiuotas rezultatas, arba pati formulė. Ką norima matyti, nurodoma meniu komandos *Tools*→*Options* skyrelyje *View* (jei norime, kad būtų matyti formulės, jungiamo jungiklį *Formulas*).

Formulę į ląstelę įrašant rankiniu būdu, reikia:

- ✚ Pažymėti ląstelę, kurioje reikia įrašyti formulę;
- ✚ Įrašyti lygybės ženklą (=);
- ✚ Įrašyti formulę;
- ✚ Paspausti <Enter> arba pažymėti kitą ląstelę.

Jeigu formulė įvesta klaidingai, programa informuoja apie klaidą. Nuspaudus mygtuką *Ok*, formulės eilutėje išskiriama klaidinga įvestos formulės dalis, tuomet būtina atlikti pataisas.

Jeigu norime naudoti funkciją, reikia:

- ✚ Pažymėti ląstelę, kurioje reikia įterpti funkciją;
- ✚ Vykdyti meniu komandą *Insert*→*Function* arba spaudžiant mygtuką  vykdyti komandą *Paste Function*;
- ✚ Pirmajame *Paste Function* dialogo lange išskirti norimą funkciją ir paspausti mygtuką *Ok*;
- ✚ Antrajame pateiktame funkcijos vykdymo lange įvesti reikalingus parametrus – skaičius, ląstelių ar sričių adresus, kitas funkcijas ir spausti mygtuką *Ok*.

Įvedus formulę nedelsiant apskaičiuojama jos reikšmė, ji pateikiama ląstelėje. Kartais *Excel* vietoj formulės reikšmės ląstelėje rodo, kad padaryta klaida (*Error value*). Programa formulėje padarytą klaidą žymi grotelėmis (#). Dažniausiai pasitaikančios klaidos:

- #NULL! – klaidingai nurodyta sritis arba sudėtinė sritis;
- #REF! – klaidingos ląstelės koordinatės;
- #VALUE! – klaidingas funkcijos argumentas ar operandas;
- #NAME? – formulėje panaudotas neatpažintas pavadinimas;
- #DIV/0! – dalyba iš nulio;

#N/A – formulėje nurodytos reikšmės neprieinamos;


#NUM! – formulėje neteisingai panaudotas skaičius.


Klaidų priežastys plačiau paaiškintos 4.9 skyriuje.

4.5.4. Ląstelių masyvas

Masyvas naudojamas tada, kai reikia sudaryti formules, kurios pateikia rezultatų rinkinį arba kurių argumentai nurodomi reikšmių rinkiniu.

Masyvai gali būti dviejų tipų:

 Ląstelių masyvas (*Array range*) yra stačiakampis ląstelių blokas;

 Konstantų masyvas (*Array constant*) yra konstantų grupė, nurodoma kaip formulės argumentas.

Rašydami formulę ląstelių masyvą įvedame pele žymėdami lentelės fragmentą, kuriame yra reikiami duomenys, pavyzdžiui, įrašydami jo koordinatas C2:D8.

Užrašydami konstantų masyvą, jo reikšmes apskliaudžiame figūriniais skliaustais, tos pačios eilutės reikšmes atskiriame simboliu \ , o eilutes – kabliataškiu, pavyzdžiui, {6\4\8;13\5\17}.

4.4 lentelė. Duomenų ląstelių masyvų pavyzdžiai

	A	B	C	D	E
1	Veikla	I ketv. Lt	II ketv. Lt	III ketv. Lt	Atskaitymai
2	1. Algos	1316,00	3364,94	6621,85	
3	2. Duomenų pirkimas	5000,00	12784,72	25159,00	9994,60
3	3. Duomenų formatavimas	4300,00	10994,86	21636,74	8791,44
5	4. Duomenų siuntimas	1060,00	2710,36	5333,71	2167,19
6	5. Informacijos parengimas	500,00	2710,36	2515,90	1022,26
7	6. Vietiniai susitikimai	1000,00	2556,94	5031,80	2044,52
8	7. Papildomos išlaidos	658,80	1684,51	3314,95	
9					
10	IŠ VISO:	13834,80	36806,70	69613,95	24020,01
11					

Masyvo formulė – tai formulė, kurios rezultatai pateikiami ląstelių masyve.

Norint surinkti masyvo formulę, reikia:

- ✚ Pažymėti ląstelių bloką, kuriame turi būti įrašyti formulės rezultatai, pavyžžiui, D2:D5;
- ✚ Surinkti formulę, ją pradant formulės redagavimo ženklu = B2:B5*C2:C5;
- ✚ Paspausti klavišus <Ctrl>+<Shift>+<Enter>.

	A	B	C	D	E
1	<i>Knygos pavadinimas</i>	<i>Kiekis</i>	<i>Kaina</i>	<i>Suma</i>	
2	Informatika	12	10	120	
3	Informacinės sistemos	5	20	100	
4	Informacinių sistemų projektavimas	7	25	175	
5	Informacinės technologijos	9	18	162	
6					

4.7 pav. Masyvo formulės panaudojimas skaičiuojant sumą

Įvesta formulė bus apskliausta figūriniais skliaustais, rodančiais, kad tai masyvo formulė. Masyvo formulėje (stulpelyje *Suma*) įrašyto ląstelių turinio keisti negalima.

Ląstelių masyvų žymėjimo pavyzdžiai pateikti 4.5 lentelėje.

4.5 lentelė. Ląstelių masyvų žymėjimo pavyzdžiai

A5:A11	Srities, apimančios septynias A stulpelio ląsteles, esančias 5–11 eilutėse (A5, A6, A7, A8, A9, A10, A11), santykinės koordinatės
B5:D6	Srities, apimančios šešias ląsteles, esančias 5–6 eilutėse ir B, C, D stulpeliuose (B5, C5, D5, B6, C6, D6), santykinės koordinatės
A2, B6:C7, D2	Sudėtinės srities, sudarytos iš ląstelių A2, B6, B7, C6, C7 ir D2, santykinės koordinatės

4.5.5. Elektroninės lentelės redagavimas ir formatavimas

Naujų stulpelių ir eilučių įterpimas

Norint įterpti naują eilutę reikia pažymėti eilutę, virš kurios reikia įterpti naują eilutę, ir vykdyti komandą *Insert*→*Rows*.

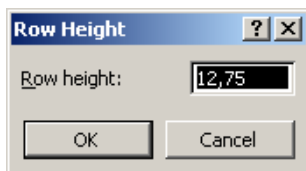
Norint įterpti naują stulpelį reikia pažymėti stulpelį, kurio kairėje reikia įterpti naują stulpelį, ir vykdyti komandą *Insert*→*Columns*.

Jeigu vienu metu norite įterpti keletą eilučių ar stulpelių, patartina žymėti tiek eilučių (stulpelių), kiek reikia įterpti.

Norint naikinti eilutes arba stulpelius, reikia pažymėti naikinamas eilutes arba stulpelius ir vykdyti komandą *Edit*→*Delete*.

Norint pakeisti eilutės aukštį ar stulpelio plotį reikia nustatyti pelės rodyklę ties eilučių, stulpelių antraštėmis ant apatinės keičiamos eilutės linijos arba į dešinę nuo pažymėtos keičiamo stulpelio vietos (pelės rodyklė įgauna kryžiaus su dviem rodyklėmis pavidalą), paspausti pelės klavišą ir laikant jį nuspaustą traukti pelės rodyklę, kol eilutė arba stulpelis taps reikiamo dydžio.

Norint pakeisti eilutės aukštį reikia pažymėti eilutę ir vykdyti komandą *Format*→*Row*, dialogo langelyje *Row Height* įrašyti norimą aukštį ir paspausti *Ok*. Šios komandos dialogo lange eilutės aukštis nurodomas punktais. Vienas punktas lygus apie 0,35 mm.



4.8 pav. Eilutės aukščio parinkimo *Format*→*Row* komandos langas

Norint pakeisti stulpelio plotį reikia vykdyti pagrindinio meniu komandą *Format*:

- ✚ Pažymėti stulpelį, kurio plotį reikia pakeisti.
- ✚ *Format*→*Column* dialogo langelyje *Column Width* įrašyti norimą plotį. Šis plotis nurodomas 10 punktų dydžio simbolių skaičiumi. Maksimalus stulpelio plotis – 255 simbolių plotis – apie 2 mm.

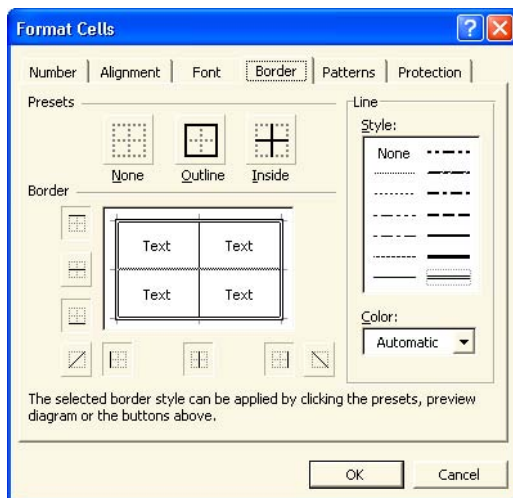


4.9 pav. Stulpelio pločio parinkimo
Format→*Column* komandos langas

Ląstelės ir srities rėmeliai

Norint nubrėžti arba pašalinti jau nubrėžtus rėmelius, reikia:

- + Pažymėti ląstelę ar sritį, kurią reikia apibrėžti rėmeliu;
- + Vykdyti komandą *Format*→*Cells* ir išrinkti lango skyrelį *Border*.



4.10 pav. Rėmelių parinkimo priemonės

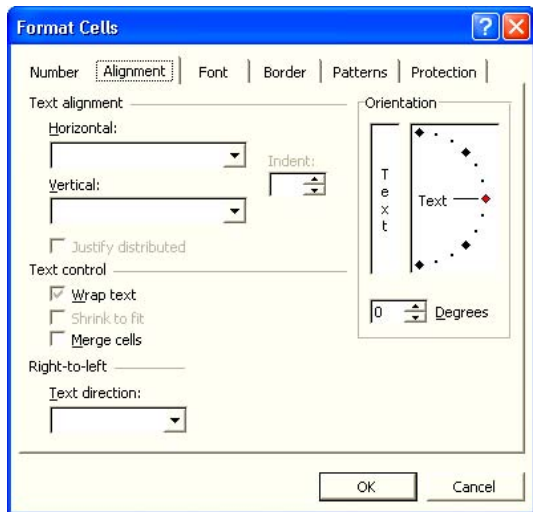
- + Išsirinkti linijos tipą (*Style*) ir spalvą (*Color*).

Lango skyrelyje *Border* nurodyti, kur brėžti rėmelius: *Outline* – aplink sritį, *Inside* – srities viduje, *Left* – kairėje, *Right* – dešinėje, *Top* – viršuje ir t. t.

Duomenų išdėstymas ląstelėse

Į ląstelę įvesti skaičiai išlyginami pagal dešiniąjį kraštą, o tekstas – pagal kairįjį.

Norint numatyta tvarka išdėstyti ląstelėse esančius duomenis, reikia pažymėti ląstelę arba sritį ir vykdant komandą *Format*→*Cells* išrinkti lango skyrelį *Alignment*.



4.11 pav. Teksto išdėstymo parinkimo priemonės

Duomenų išdėstymo ląstelėje būdai:

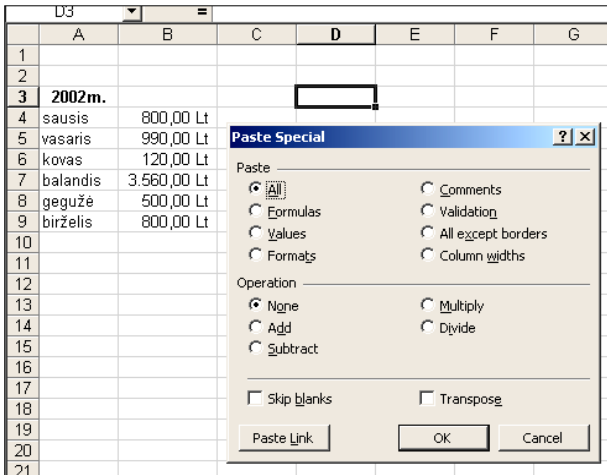
- ✚ *Horizontal* – centruoti horizontalia kryptimi;
- ✚ *Vertical* – centruoti vertikalia kryptimi;
- ✚ *Orientation* – keisti teksto kryptį;
- ✚ *Wrap text* – tekstą, esantį ląstelėje, išdalyti į eilutes;
- ✚ *Merge cells* – sujungti ląsteles;
- ✚ *Shrink to fit* – užpildyti ląstelę joje įrašyta reikšme.

4.5.6. Specialūs duomenų kopijavimo būdai

Norint kopijuoti arba perkelti tik dalį informacijos, esančios ląstelėje arba srityje, reikia tai atlikti tam tikru būdu:

- ✚ Pažymėti kopijuojamą arba perkeliamą ląstelę arba sritį;

- ✚ Norint kopijuoti reikia vykdyti komandą *Edit*→*Copy*, norint perkelti – *Edit*→*Cut*. Kopijuojama, perkeliama dalis apibrėžiama mirksinčia punktyrine linija.
- ✚ Pažymėti laštelę (arba srities kairiąją viršutinę laštelę), į kurią reikia kopijuoti (perkelti) duomenis.
- ✚ Vykdyti komandą *Edit* → *Paste Special*;



4.12 pav. *Edit*→*Paste Special* komandos langas, skirtas kopijuoti ar perkelti informaciją, renkantis parametru reikšmes

Komandos dialogo lange *Paste Special* reikia pasirinkti vieną iš komandos parametru.

4.6 lentelė. Kopijavimo komandos *Paste Special* parametru reikšmės


<i>All</i>	Įterpti visą laštelių informaciją (formules, reikšmes, formatus, komentarus)
<i>Formulas</i>	Įterpti tik formules
<i>Values</i>	Įterpti tik reikšmes
<i>Formats</i>	Įterpti tik formatavimo parametrus
<i>Comments</i>	Įterpti tik komentarus
<i>Alle except borders</i>	Įterpti visą informaciją, išskyrus rėmelius
<i>Column Widths</i>	Įterpti stulpelių pločių reikšmės

4.7 lentelė. Kopijavimo komandos *Paste Special* operacijų atlikimo dalies *Operation* parametrų reikšmės

<i>None</i>	Kopijuojamos (perkeliamos srities) duomenis perrašyti į įterpimo sritį
<i>Add</i>	Prie įterpimo srities ląstelių reikšmių pridėti kopijuojamos (perkeliamos) srities duomenis
<i>Subtract</i>	Iš įterpimo srities ląstelių reikšmių atimti kopijuojamos (perkeliamos) srities duomenis
<i>Multiply</i>	Įterpimo srities ląstelių reikšmės padauginti iš kopijuojamos (perkeliamos) srities duomenų
<i>Divide</i>	Įterpimo srities ląstelių reikšmės padalinti iš kopijuojamos (perkeliamos) srities duomenų

4.5.7. Darbo pabaigos komandos

Galimi keli programos pabaigos būdai:

1. Pele paspausti lango uždarymo mygtuką *Close* ;
2. Paspausti klavišus *<Alt>+<F4>*;
3. Vykdyti komandą *File→Exit*.

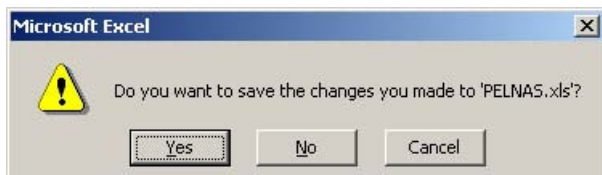
Jeigu dokumentas buvo redaguojamas, bet pakeitimai nebuvo įrašyti į failą, baigiant dirbti programa pasirodo užklausa *Do you want to save the changes you made to 'dokumento vardas'*, patikslinanti, kaip norima išsaugoti darbo knygą.

Tokiu atveju reikia pasirinkti vieną iš pateiktų atsakymo variantų:

Yes – išsaugoti dokumento pakeitimus;

No – pakeitimų neišsaugoti;

Cancel – tęsti darbą, atšaukiant komandą.



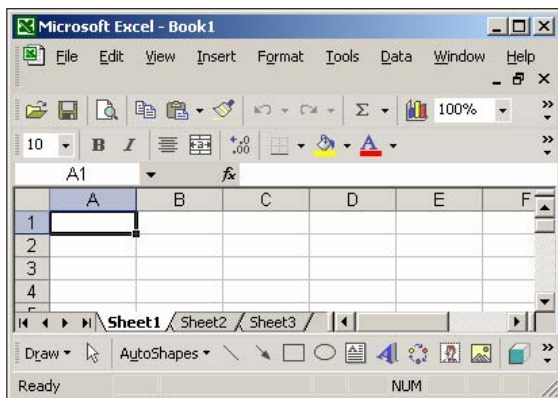
4.13 pav. Darbo pabaigos informacinis pranešimas, kai dokumentas redaguojamas ir pakeitimai išsaugomi

Baigiant dirbti su programa, tie patys pranešimai bus matyti asistento langelyje, jeigu jis įjungtas (t. y. vykdyta komanda *Help*→*Show the Office Assistant*).

4.5.8. Lentelių rinkinio puslapių formavimas

Naujai atidaryta *Excel* knyga turi 3 lapus (*Sheet1*, *Sheet2*, *Sheet3*).

Norint pereiti nuo vieno lapo prie kito reikia spragtelėti pele ties reikiamo lapo pavadinimu lapų sąrašo eilutėje, pavyzdžiui, *Sheet2*.



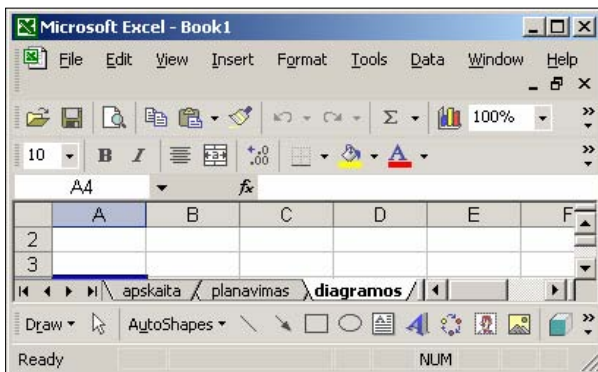
4.14 pav. Pagrindinis *Excel* programos langas esant aktyviam pirmam lapui

Norint suaktyvinti kelis lapus iš karto, reikia paspausti klavišą *< Ctrl >*, spragtelėti pele ties reikiamų lapų pavadinimais.

Darbo lapo pavadinimo keitimas

Excel knygoje darbo lapai vadinami *Sheet1*, *Sheet2* ir t. t. Knygą bus lengviau skaityti, jei lapai turės prasminius pavadinimus, pavyzdžiui, *apskaita*, *planavimas*, *diagramos*, ir pan.

Norint suteikti lapui kitą pavadinimą, kuris nusakytų jame saugomos informacijos prasmę ir leistų lengviau susirasti duomenis didelės apimties darbo knygoje, reikia dešiniuuoju pelės klavišu spragtelėti ties pasirinktojo lapo pavadinimu, pavyzdžiui, *Sheet1*, iškviešti kontekstinį meniu ir vykdyti komandą *Rename*; klaviatūra surinkti kitą lapo pavadinimą.

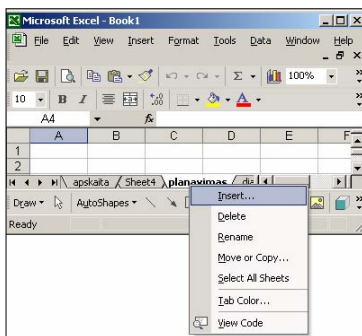


4.15 pav. Excel programos langas suteikus darbo lapams naujus pavadinimus

Naujo lapo įterpimas

Norint įterpti naują lapą reikia:

- ✚ Pažymėti lapą, prieš kurį reikia įterpti naują lapą;
- ✚ Iškvieisti kontekstinį meniu ir vykdyti komandą *Insert*;
- ✚ Naujas lapas įterpiamas į kairę nuo pasirinktojo.



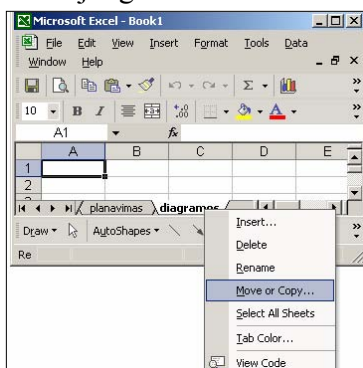
4.16 pav. Naujo lapo įterpimo komanda, iškviečius kontekstinį meniu

Darbo lapų kopijavimas ir perkėlimas

Norint nukopijuoti ar perkelti darbo lapą (lapus) į kitą arba tą pačią knygą reikia:

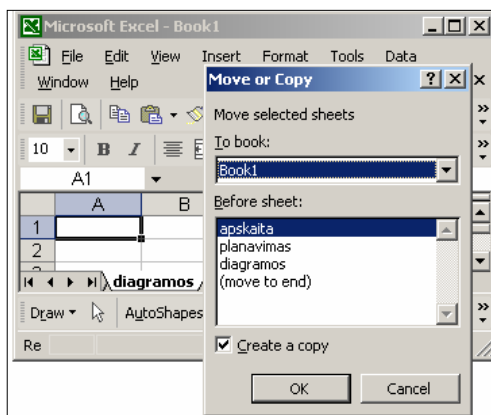
- ✚ Pažymėti kopijuojamą ar perkeliamą lapą;
- ✚ Iškvieisti kontekstinį meniu ir vykdyti komandą *Move or Copy*;

- ✚ Komandos dialogo lange *Move or Copy* sąrašė *To Book* išrinkti knygą, į kurią bus kopijuojamas arba perkeliamas lapas;
- ✚ Sąrašė *Before Sheet* pažymėti lapą, prieš kurį bus įterpta (perkeliama) kopija;
- ✚ Kopijuojant lapą, reikia įjungti *Create a Copy*, perkeliant lapą jungiklis neįjungiamas.



4.17 pav. Aktyvaus lapo perkėlimo ar kopijavimo komanda iškvietus kontekstinį meniu

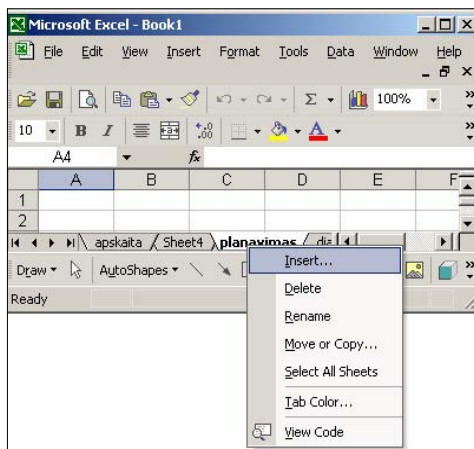
Kopijuojant lapą, būsiantį toje pačioje darbo knygoje, prie jo pavadinimo prirašomas skaičius, pavyzdžiui, *diagramos(2)*.



4.18 pav. Kopijuojamo ar perkeliama lapo vietos parinkimo langas

Lapo panaikinimas

Norint naikinti nebereikalingus ar perteklinius darbo knygos lapus, reikia pažymėti lapą ar kelis lapus, kuriuos reikia panaikinti; iškviešti pažymėtojo lapo kontekstinį meniu ir vykdyti komandą *Delete*.



4.19 pav. Kontekstinis planavimo lapo meniu

4.6. FORMULIŲ SUDARYMAS SKAIČIUOKLĖSE




Formulės skirtos atlikti skaičiavimus lentelėse. Pagal formulėje nurodytas reikšmes apskaičiuojama nauja reikšmė. Sudarant formulę ląstelėje, kurioje turės būti įrašytas reikalingas rezultatas, įrašomas formulės pradžios ženklas – lygybės simbolis =. Formulėje nurodytos reikšmės gali būti konstantos, įrašytos pačioje formulėje, arba reikšmės, įrašytos ląstelėse, kurių adresai nurodyti formulėje. Formulėje įrašytos operacijos ir funkcijos apibrėžia, kokius veiksmus reikia atlikti. Sudarant formulę ląstelių ar masyvų koordinatas galima įterpti automatiškai, pele pažymint reikiamą ląstelę arba jų masyvą.

Formulėse naudojami aritmetiniai, teksto jungimo ir santykio operatoriai (4.8 lentelė).

4.8 lentelė. Operacijų sąrašas

Operacijų eilės tvarka	Operacijos ženklas	Operacija
1	-	Skaičiaus ženklo keitimas
2	%	Procentų apskaičiavimas
3	^	Kėlimas laipsniu
4	*	Daugyba
	/	Dalyba
5	+	Sudėtis
	-	Atimtis
6	&	Teksto fragmentų jungimas
7	<; <=	Mažiau; mažiau arba lygu
	>; >=	Daugiau; daugiau arba lygu
	=	Lygu
	<>	Nelygu

Skaičiuojant rezultatus formulėse:

-  Pirmiausia apskaičiuojami skliaustuose esantys reiškiniai, tada atliekamos operacijos su skliaustuose esančiomis reikšmėmis;
-  Veiksmai atliekami pagal operacijų (pirmenybės) eilės tvarką.
-  Kai operacijos tos pačios eilės, veiksmai atliekami iš kairės į dešinę.

Formulės sudedamosios dalys yra operacijų ženklai, ląstelių koordinatės, skaičių konstantos ir funkcijos.

Formulių pavyzdžiai:

=B5+C10*2;
 =(B5+C10-\$D\$10)/4;
 =(B5+C10)*33%;
 =B\$5+AVERAGE(C5:C10).

Įrašius formules, pavyzdžiui, į D stulpelį, atitinkamoje eilutėje gaunami skaičiavimo pagal sudarytąsias formules rezultatai.

	A	B	C	D	E
1					
2	Sudėtis	10	4	14	=B2+C2
3	Atimtis	80	12	68	=B3-C3
4	Sandauga	3	10	30	=B4*C4
5	Dalyba	100	4	25	=B5/C5
6	Kėlimas laipsniu		2	4	=C6^2
7	Sujungimas	Teisės	aktai	Teisės aktai	=B7&C7
8	Skiaustai			602	=(C6+C6)*100+2
9	Funkcijos panaudojimas			129,5	=AVERAGE(B2:C3)+SUM(B4:B5)
10					

4.20 pav. Formulių sudarymo pavyzdžiai

Formulės redagavimas

Formules patogų redaguoti formulės eilutėje. Pažymėjus ląstelę, kurioje yra formulės rezultatas, formulės eilutėje pateikiama pati formulė. Elektroninės lentelės leidžia redaguoti formules pačioje ląstelėje. Kairiuoju pelės klavišu du kartus spragtelėjus redaguojamą ląstelę, formulė rodoma ląstelėje. Redaguoti baigiama, spaudžiant <Enter> klavišą arba jį atitinkantį mygtuką (paukščiuką) formulės eilutėje.

Jeigu reikia, kad lentelėje vienu metu būtų matyti visos formulės, o ne apskaičiuoti rezultatai, tai padaryti galima įjungiant formulių peržiūros režimą. Tuomet reikia vykdyti komandą *Tools* → *Options* bei šios komandos dialogo lango skyrelyje *View* įjungti *Formulas*.

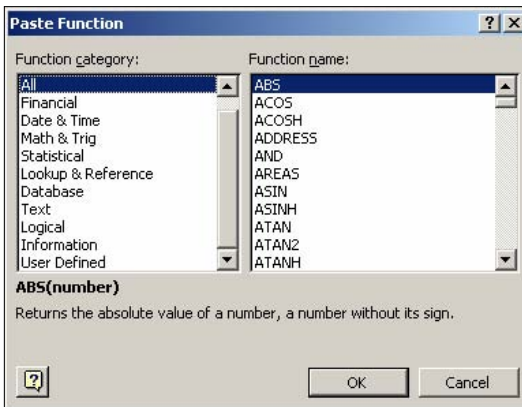
4.7. FUNKCIJŲ PANAUDOJIMAS

Pasitelkdami funkcijas sparčiau sudarysime elektronines lenteles įvairiems skaičiavimams atlikti. Funkcijos atlieka dažnai naudojamas matematinių veiksmų, teksto, statistinio duomenų apdorojimo bei finansinių skaičiavimų sekas ir kitus veiksmus.

Skaičiuoklės turi be galo daug funkcijų, kurios, kad vartotojams būtų patogiau, suskirstytos į grupes:

- Visos funkcijos, surašytos pagal abėcėlę;

- ✚ Finansinės funkcijos;
- ✚ Datos ir laiko funkcijos;
- ✚ Matematinės ir trigonometrinės funkcijos;
- ✚ Statistinės funkcijos;
- ✚ paieškos ir adresų funkcijos;
- ✚ Duomenų bazės funkcijos;
- ✚ Teksto apdorojimo funkcijos;
- ✚ Loginės funkcijos;
- ✚ Informacinės funkcijos;
- ✚ Vartotojo funkcijos.



4.21 pav. Excel programos funkcijų grupės su aktyviu visų funkcijų sąrašu

Funkciją sudaro pavadinimas ir nustatyta tvarka skliaustuose surašyti privalomi bei neprivalomi argumentai, atskiriami kabliataškiais, kableliais bei dvitaškiais.

Bendroji funkcijos užrašymo forma:

Funkcijos pavadinimas (a1; a2; a3;...), a1, a2, a3 – funkcijos argumentai.

Funkcijos pavadinimas nurodo funkcijos paskirtį:

SUM – sumavimo funkcija;

AVERAGE – vidurkio skaičiavimo funkcija;

MAX – maksimumo – didžiausios reikšmės pateikimo funkcija;

MIN – minimumo – mažiausios reikšmės pateikimo funkcija ir panašiai.

Kiekviena funkcija gali turėti nuo 0 iki 30 argumentų. Argumentai gali būti visų šešių *Excel* programoje naudojamų duomenų tipų:

- ✚ Skaičiai, pavyzdžiui, 7; -8; 5,4; 2,5;
- ✚ Teksto eilutės, pavyzdžiui, *a*; *Pranešimas*;
- ✚ Loginės reikšmės, pavyzdžiui, *True* arba *False*;
- ✚ Koordinatės, pavyzdžiui, A7:A8; G5; \$F\$4;
- ✚ Masyvai, pavyzdžiui, {1\2\3; 5\6\7};
- ✚ Klaidos reikšmės, pavyzdžiui, #N/A; #NUM!; #VALUE!

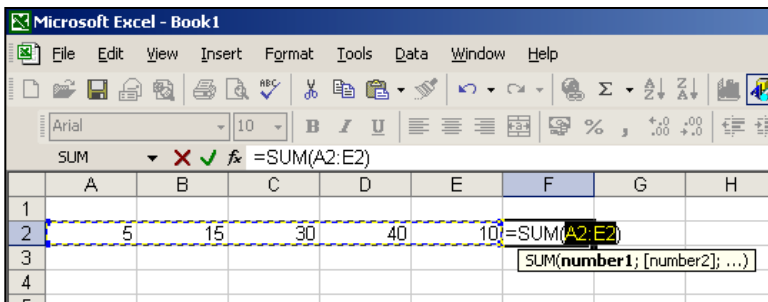
Funkcijos argumentas gali būti kita funkcija. *AVERAGE* funkcija gali būti *SUM(A2:A8; AVERAGE(5; 9))*, tuomet *AVERAGE* yra pirmojo lygmens įdėtinė funkcija. Vienoje formulėje tokių lygių skaičius gali būti lygus septyniems, tačiau pačios formulės ilgis negali turėti daugiau nei 1024 simbolių.

4.7.1. Matematinės ir trigonometrinės funkcijos

Sumavimo funkcija

Norint susumuoti skaičius, taikoma sumavimo funkcija *SUM* (skaičius 1; skaičius 2, ...). Sudarant šią funkciją galima naudoti iki trijų dešimčių argumentų. Pavyzdžiui, *SUM(3;2)* rezultatas – 5.

Jeigu į ląsteles nuo A2 iki E2 įrašyti skaičiai 5, 15, 30, 40, 10, funkcijos argumentą galima pateikti kaip ląstelių bloką, pavyzdžiui, *SUM(A2:E2)* rezultatas bus 100. Pateikta funkcija *SUM(A2:E2)* yra ekvivalenti *SUM(A2;B2;C2;D2;E2)*, tačiau pirmoji yra pranašesnė, nes turi tik vieną argumentą, o antroji – 5.



4.22 pav. Sumavimo funkcijos pavyzdys

Funkcija $SUM(A:A)$ – funkcijos argumentų adresuose nurodžius tik stulpelio adresą pažymima, kad bus sumuojami visi nurodyto stulpelio skaičiai.

	A	B	C	D
1				
2	30	40	10	80
3	5			
4	60			
5	90			
6	100			
7	90			
8	80			
9	170	=SUM(A:A)		
10				
11				
12				
13				

4.23 pav. Sumavimo funkcijos pavyzdys, sumuojant visus nurodyto stulpelio skaičius

$SUM(2:2)$ – sumavimo funkcijos argumentų adresuose nurodant tik eilutės adresą pažymima, kad bus sumuojami visi nurodytosios eilutės skaičiai.

Skaičių apvalinimo funkcijos

	A	B	C	D	E	F	G
1							
2	30,3388	=ROUND(A2;2)					
3	5,5699						
4	60,6699						
5	90,5897						

Function Arguments

ROUND

Number: A2 = 30,3388

Num_digits: 2 = 2

Rounds a number to a specified number of digits. = 30,34

Num_digits is the number of digits to which you want to round. Negative rounds to the left of the decimal point; zero to the nearest integer.

Formula result = 30,34

[Help on this function](#) [OK] [Cancel]

4.24 pav. Skaičių apvalinimo funkcijos pavyzdys

Funkcija ROUND(skaičius;ženklų skaičius po kablelio) apvalina skaičius, palikdama nurodytą ženklų skaičių po kablelio.

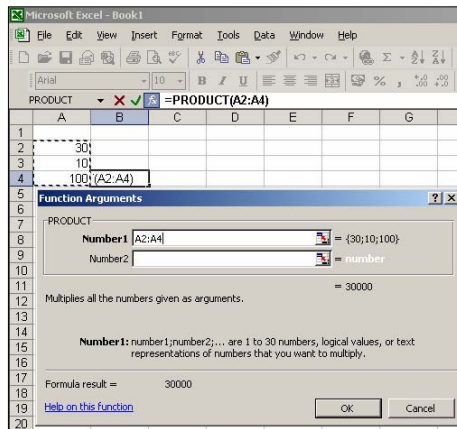
ROUNDUP(skaičius;ženklų skaičius po kablelio) – apvalinimas prie paskutinio skaitmens pridėdam vienetą.

ROUNDDOWN(skaičius;ženklų skaičius po kablelio) – apvalinimas nekeičiant paskutinio skaitmens.

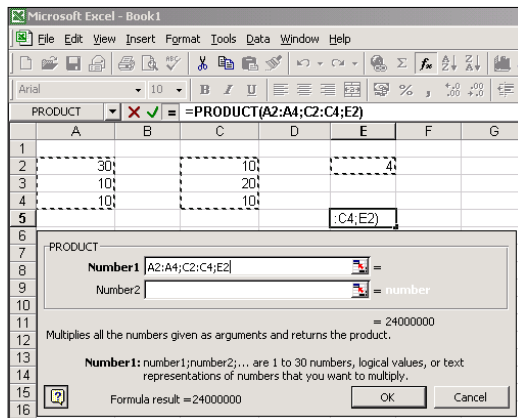
Sandaugos funkcija

PRODUCT(skaičius1;skaičius2;...) skirta daugybos veiksmams atlikti.

Pasitelkiant šią funkciją galima sudauginti iki 30 argumentų.



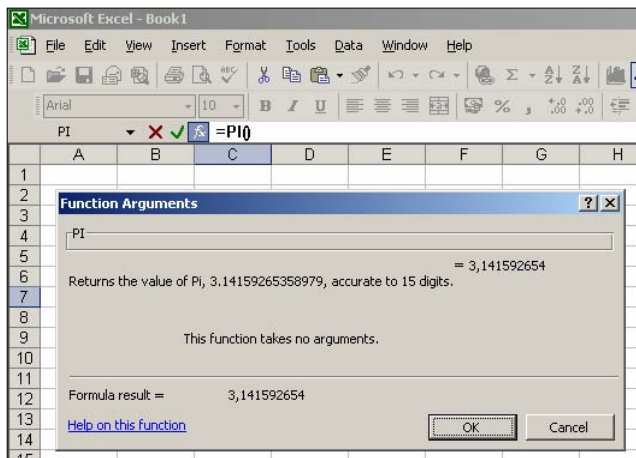
4.25 pav. Sandaugos funkcijos pavyzdys



4.26 Sandaugos funkcija naudojant kelis argumentus

Skaičiaus π funkcija

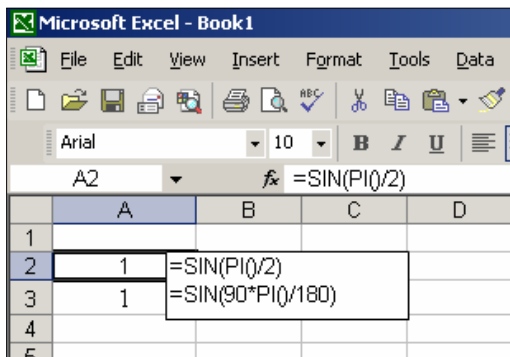
Naudojant trigonometrines funkcijas jų argumentas dažnai pateikiamas kaip skaičius π . Šio skaičiaus reikšmė – 3,14159. Skaičių π programa išreiškia 15 ženklų po kablelio tikslumu kaip funkciją $PI()$.



4.27 pav. Skaičiaus π funkcija ir jos naudojimo būdas

Trigonometrinės funkcijos

Trigonometrinė funkcija SIN (skaičius) apskaičiuoja kampo sinusą. Jos argumentas gali būti nurodomas radianais, pavyzdžiui, $=SIN(PI()/2)$, arba laipsniais $=SIN(90*PI()/180)$.



4.28 pav. Trigonometrinės funkcijos kampo sinuso argumentų pavyzdžiai

Trigonometrinė funkcija COS (skaičius) apskaičiuoja kampo kosinusą, o funkcija TAN (skaičius) – tangentą.

Excel programa analogiškai skaičiuoja funkcijas $ARCSIN$, $ARCCOS$ ir $ARCTAN$.

Funkcija $SQRT$ (skaičius) skirta kvadratinei šakniai skaičiuoti. Pavyzdžiui, $SQRT(16)$ rezultatas bus 4.

Funkcija $FACT$ (skaičius) skirta faktorialui skaičiuoti. Pavyzdžiui, $FACT(7)$ rezultatas bus 5040 ($1*2*3*4*5*6*7=5040$).

4.7.2. Statistinės funkcijos

Vidurkio skaičiavimo funkcija

Funkcija $AVERAGE$ (skaičius1;skaičius2;...) nustato pateiktų skaičių aritmetinį vidurkį. Šiai funkcijai sudaryti galima naudoti iki trijų dešimčių argumentų. Ji apskaičiuoja skaičių arba ląstelių adresuose esančių skaičių aritmetinį vidurkį.

Minimalios reikšmės išrinkimo statistinė funkcija

Funkcija MIN (skaičius1;skaičius2;...) nustato minimalų iš pateiktų skaičių. Šiai funkcijai sudaryti galima naudoti iki trijų dešimčių argumentų.

Ji suranda skaičių arba mažiausią ląstelių adresuose esančių skaičių reikšmę.

Maksimalios reikšmės pateikimo funkcija

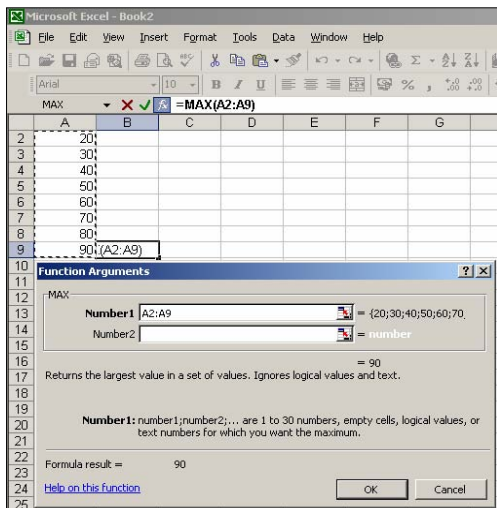
Funkcija MAX (skaičius1;skaičius2; ...) nustato maksimalų skaičių. Šiai funkcijai sudaryti galima naudoti iki trijų dešimčių argumentų.

Ši funkcija suranda skaičių arba ląstelių adresuose esančių skaičių didžiausią reikšmę.

Ląstelių reikšmių analizės funkcijos

$COUNT$ (reikšmė;reikšmė;...) funkcija suskaičiuoja, kiek skaičių yra koordinatėmis nurodytame bloke.

$COUNTA$ (reikšmė;reikšmė;...) – įveda įrašytų argumentų skaičių, ją naudojame norėdami suskaičiuoti visas netuščias ląsteles.



4.29 pav. Maksimalios reikšmės pateikimo funkcijos naudojimo būdas

4.7.3. Finansinės funkcijos

Funkcija $FV(rate;nper;pmt;pv;type)$ apskaičiuoja būsimąją investicijos vertę. Funkcijos parametrai:

Rate – norma, pavyzdžiui, 7 proc. palūkanų;

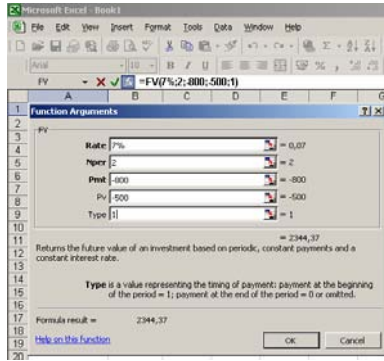
Nper – periodų skaičius, pavyzdžiui, jeigu mokėdami kasmet gautą paskolą turite grąžinti per 2 metus, $nper = 2$;

Pmt – vienodų periodinių mokėjimų dydis, pavyzdžiui, kasmet mokate po 800 Lt;

Pv – esamoji vertė, pavyzdžiui, pasiskolintų pinigų suma – 500 Lt;

Type – požymis 0, kai mokama periodo gale, 1 – kai pradžioje.

Norint apskaičiuoti investicijos esant nurodytoms sąlygoms vertę, pasitelkiama FV funkcija. Esant nurodytoms sąlygoms investicijos vertė sudarys 2344,37 litų.



4.30 pav. Investicijos vertės apskaičiavimo funkcijos argumentų ir rezultato pavyzdys

Funkcija $DB(\text{Cost};\text{Salvage};\text{Life};\text{Period};\text{Month})$ fiksuoto mažėjančio likučio metodu suskaičiuoja turto nusidėvėjimą per tam tikrą laikotarpį. Šios funkcijos parametrai:

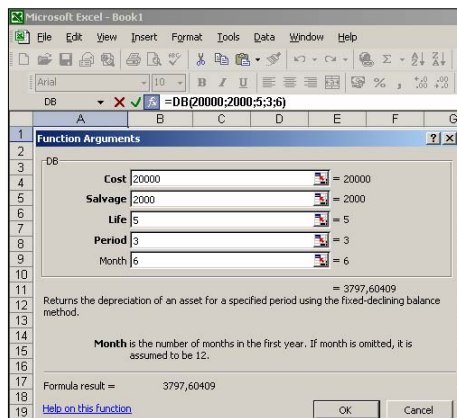
Cost – pradinė turto kaina, pavyzdžiui, 20000;

Salvage – liekamoji vertė baigus naudoti turta, pavyzdžiui, 2000;

Life – naudojimo trukmė, pavyzdžiui, periodų skaičius, kuriems praėjus turtas nusidėvi iki liekamosios vertės, pavyzdžiui, 5 metai;

Period – periodų skaičius, pavyzdžiui, po 3 metų;

Month – mėnesių skaičius pirmaisiais metais, pavyzdžiui, 6.



4.31 pav. Turto nusidėvėjimo per tam tikrą periodą (skaičiuojant fiksuoto mažėjančio likučio metodu) funkcijos argumentų ir jos rezultato pavyzdys

Pradinė turto kaina – 20000 litų, liekamoji vertė – 2000 litų, iki turto susidėvėjimo liko 5 metai, norime sužinoti turto nusidėvėjimo vertę po 3 metų.

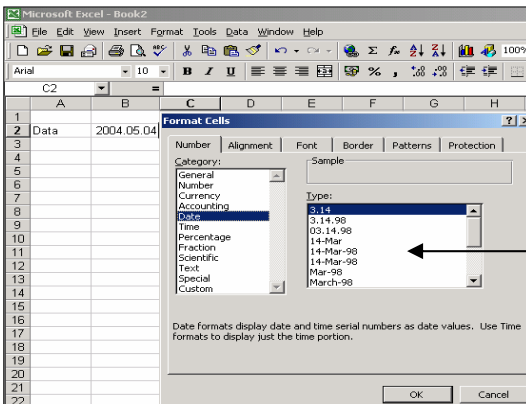
Pasitelkdami *DB* funkciją esant nurodytoms sąlygoms apskaičiavome, kad turtas per trejus metus nuvertėjo 3707,06 litų.

4.7.4. Laiko apskaitos funkcijos

Laikas šiuolaikiniame gyvenime tampa labai svarbus veiksnys, o kartais ir vienas pagrindinių sprendimų priėmimo kriterijų. Todėl labai svarbu turėti galimybių atlikti laiko apskaitą ir tinkamai naudotis siūlomomis laiko parametrų įvertinimo priemonėmis.

Datos skaičiuoklėse numeruojamos skaičiais nuo 1 iki 2 958 465. Pirmoji apskaitos data yra 1900 metų sausio 1 diena, jai suteiktas numeris – 1. Didžiausią numerį atitinka 9999 metų gruodžio 31 diena. Konkreti data traktuojama kaip sveikasis skaičius, didesnis už 1 ir mažesnis už 2958465.

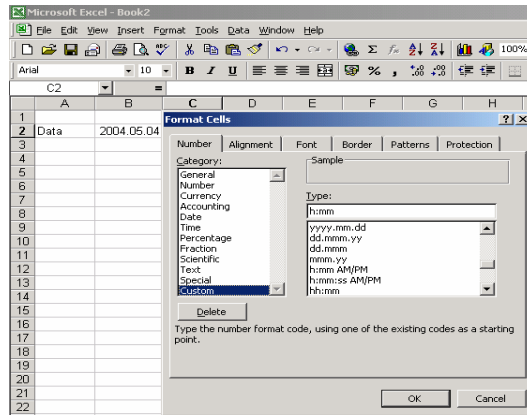
Toks datų apskaitos būdas leidžia atlikti aritmetinius veiksmus su datomis, lengvai keisti datų pateikimo būdą, apskaičiuoti laiko trukmę ir panašiai. Ląstelėje įrašytam duomeniui datos formatą galima suteikti *Format*→*Cells*→*Number* komanda. Pasirinkdami datos (*Date*) arba laiko (*Time*) duomenų tipą, norimą pateikimo būdą šiems duomenims suteiksite pasirinkdami vieną variantą iš duomenų formatų sąrašo.



4.32 pav. Datų formatų sąrašas

Norint sukurti savą datos ar laiko formatą, reikia pažymėti vartotojui skirtą kategorijų grupę *Custom* ir iš jos pateikiamo sąrašo pasirinkti pageidaujama formatą. Nusakant datų formatus taikomi kai kurie tam tikrą reikšmę turintys simboliai:

- ✚ Dienai nurodyti dviejų skaičių junginiu pažymima raidžių grupė *dd*;
- ✚ Dienai nurodyti dviejų lietuviškų raidžių junginiu, nurodančiu savaitės dienas, – *ddd*, pavyzdžiui, Pr, An ir t. t.
- ✚ Pateikiant visą mėnesio pavadinimą – *mmmm*;
- ✚ Metų numeriui išreikšti dviženkliais skaičiais – *yy* arba keturženkliais skaičiais – *yyyy* ir kt.



4.33 pav. Individualaus datos formato kūrimo būdas

Lietuviškai išreiškiamą datų užrašymo tvarką patogiausia pasirinkti per operacinės sistemos *Windows Control Panel* valdymo skydelio šalies parametrus *Regional Settings Date*, esančius formato nustatymo lange, pavyzdžiui, 2004.05.04.

Darbo lapų ląstelėse įrašytą datą ir laiką galima naudoti sudarant formules ir funkcijas, nustatant einamąjį kompiuterio datą, atliekant įvairius veiksmus su datomis.

Atliekant aritmetinius veiksmus su datomis ir laiku, pirmiausia reikia žinoti vieną svarbią taisyklę: neatsižvelgiant į tai, kokių formatu celėje įrašyta data, viena para datų numerių sekoje visada lygi

vienetui, o šio vieneto trupmeninė dalis atitinka paros laiką, t. y. valandas, minutes, sekundes.

Konkrečios datų ir laiko funkcijos apskaičiuoja ir pateikia įvairias tokios datos arba argumentais nurodyto laikotarpio dalis.

Skaičiuoklė turi apie 20 datų ir laiko funkcijų [26]. Pateiksime bendrus dažniausiai naudojamų datų ir laiko funkcijų aprašus ir paaiškinsime jų paskirtį:

DATE(year;month;day) – funkcija, pateikianti nurodytos datos eilės numerį;

DATEVALUE(date_text) – pateikia tekstu nurodytos datos eilės numerį;

DAYS360(start_date;end_date;method) – pateikia trukmę, nurodama dienų skaičių tarp pradinės *start_date* ir galinės *end_date* datos;

DAY(serial_number) – pateikia eilės numeriu nurodyto mėnesio dieną;

HOUR(serial_number) – pateikia eilės numeriu nurodytos datos valandas;

NETWORKDAYS(start_date;end_date;holidays) – pateikia darbo dienų skaičių tarp dviejų datų. Darbo dienomis nelaikomi savaitgaliai ir pažymėtos šventės;

NOW() – pateikia einamąją datą ir laiką;

TIME(hour;minute;second) – paverčia nurodytą laiką eilės numeriu;

TODAY() – pateikia einamąją datą;

WEEKDAY(serial_number;return_type) – pateikia eilės numeriu *serial_number* nurodytos datos savaitės dieną, kai nurodytas dienų numeravimo tipas *return_type*.

YEAR(serial_number) – pateikia eilės numeriu nurodytos datos metus.

YEARFRAC(start_date;end_date;basis) – pateikia santykinę metų dalį (dešimtaine trupmena), atitinkančią visų dienų, esančių tarp dviejų datų – *start_date* ir *end_date* – dalį metuose. Ši dalis gali būti skaičiuojama remiantis skirtingomis datų sistemomis (*basis*).

Skaičiuoklėse yra ir kitų laiko apskaitos funkcijų, jos plačiau apibūdintos V. Šakio knygoje „Skaičiuoklė Excel’2000 firmos vadymui“.

4.7.5. Informacinės funkcijos

Pasitelkiant informacines funkcijas galima patikrinti nurodytų ląstelių ar blokų turinį ir pateikti įvairią informaciją apie ląstelės ar bloko reikšmės tipus, formatavimo parametrus, įrašytus duomenis ir pan.

4.9 lentelė. Informacinių funkcijų pavyzdžiai

Informacinės funkcijos aprašas	Funkcijos rezultatas
<i>CELL</i> (<i>type_of_info;reference</i>)	Pateikia informaciją apie ląstelės formatą, vietą, įrašytus duomenis
<i>COUNTBLANK</i> (<i>range</i>)	Nurodo suskaičiuotas tuščias pažymėto bloko ląsteles
<i>INFO</i> (<i>type_text</i>)	Pagal teksto tipą pateikia informaciją apie atminties dydį, atidarytų bylų skaičių
<i>IS****</i> (argumentas) tipo funkcijos (yra 11 tokio tipo funkcijų)	<i>TRUE</i> , jei argumentas yra tam tikro tipo, arba <i>FALSE</i> , jei argumentas neatitinka nurodyto tipo
<i>IS BLANK</i> (<i>value</i>)	<i>TRUE</i> , jei ląstelė tuščia
<i>IS ERROR</i> (<i>value</i>)	<i>TRUE</i> , jei argumentas yra viena iš klaidos reikšmių
<i>N</i> (<i>value</i>)	Paverčia reikšmę skaičiumi
<i>N</i> ()	Pateikia klaidos reikšmę #N/A
<i>TYPE</i> (<i>value</i>)	Nurodo reikšmės tipą

4.7.6. Loginės funkcijos

Loginės funkcijos skirtos palyginti, patikrinti, sujungti argumentus ir pagal sąlygą apskaičiuoti reikšmes [26]. Jos leidžia pagal Bulio algebros taisykles įvertinti argumentų konjunkciją (*AND*), disjunkciją (*OR*), neiginį (*NOT*). *MS Excel* turi šešias logines funkcijas.

4.10 lentelė. Loginių funkcijų pavyzdžiai

Loginės funkcijos aprašas	Pateikiama reikšmė
$AND(loginis1;loginis2;...)$	$TRUE$, jei kiekvieno argumento reikšmė yra $TRUE$, $FALSE$, jei bent vieno argumento reikšmė yra $FALSE$
$OR(loginis1;loginis2;...)$	$TRUE$, jei bent viena iš argumentų reikšmių yra $TRUE$; $FALSE$, jei visos argumentų reikšmės yra $FALSE$
$NOT(logical)$	$TRUE$, jei tikrinimo sąlygos reikšmė netenkinama; $FALSE$, jei tikrinimo sąlygos reikšmė tenkinama
$TRUE()$	Pateikiama loginė reikšmė $TRUE$
$FALSE()$	Pateikiama loginė reikšmė $FALSE$
$IF(loginė_sąlyga; reikšmė_jei_loginė_sąlyga_TRUE; reikšmė_jei_loginė_sąlyga_FALSE)$	Pateikiamas vienas iš apskaičiuotų reikšmės rezultatų, priklausantis nuo įvertintos sąlygos atsakymo ($TRUE$ arba $FALSE$)

Atliekant ekonominius skaičiavimus dažniausiai naudojama loginė funkcija IF . Renkantis šią funkciją galima nurodyti sudėtingą loginę sąlygą, sudarytą iš kitų AND ir OR loginių funkcijų. IF funkcija gali turėti septynis IF lygmenis. AND ir OR loginės funkcijos gali turėti nuo vieno iki trisdešimties argumentų.

The screenshot shows a Microsoft Excel spreadsheet titled "Microsoft Excel - stipendijos.xls". The formula bar at the top displays the formula: $=IF(D5<4,0,IF(D5<6, \$E\$2,IF(D5<8, \$E\$2*1,2, \$E\$2*2)))$. The spreadsheet contains a table with the following data:

Studentų stipendijų žiniaraštis				
Stipendijos minimumas				60,00 Lt
Data 2004.02.05				
Eil. Nr.	Pavardė V.	Egzaminų balų suma	Balų vidurkis	Stipendijos dydis
1	Alutis J.	60	6	72,00 Lt
2	Augulytė R.	80	8	120,00 Lt
3	Bertauskaitė P.	100	10	120,00 Lt
4	Jonaitis V.	30	3	- Lt
5	Petraitis D.	50	5	60,00 Lt
6	Railaitė K.	70	7	72,00 Lt
7	Šalna E.	90	9	120,00 Lt
Bendra suma				564,00 Lt

4.34 pav. Loginės funkcijos IF pavyzdys

Pateiktame pavyzdyje stipendijos dydis apskaičiuojamas sudarant dviejų lygmenų *IF* funkciją.

Kiti sudėtingesni *IF* panaudojimo pavyzdžiai bus pateikti atlyginimų apskaitos praktiniuose darbuose.

4.7.7. Paieškos ir koordinacių funkcijos

Paieškos funkcijos taikomos, kai lentelėje reikia rasti nežinomas objekto charakteristikas, pavyzdžiui, masyvo eilučių skaičių, ląstelių eilučių numerius, kai duotos jų koordinatės arba kai reikia rasti masyvo elemento reikšmę.

ROW(reference) – kai nurodomos masyvo koordinatės (*reference*), funkcija pateikia ląstelių eilučių numerius;

ROWS(array) – kai nurodomos masyvo koordinatės (*array*), funkcija pateikia masyvo eilučių skaičių.

Analogiškos funkcijos:

COLUMN(reference) – kai nurodomos koordinatės, funkcija pateikia ląstelių stulpelių numerius;

COLUMNS(array) – kai nurodomos masyvo koordinatės, funkcija pateikia masyvo stulpelių skaičių (kiekį).

Masyvas naudojamas tada, kai reikia sudaryti formules, kurios pateikia rezultatų rinkinį arba kurių argumentai nurodomi reikšmių rinkiniu. Masyvai gali būti dviejų tipų: ląstelių masyvas (*array range*) yra stačiakampis ląstelių blokas, konstantų masyvas (*array constant*) yra konstantų grupė, nurodoma kaip formulės argumentas.

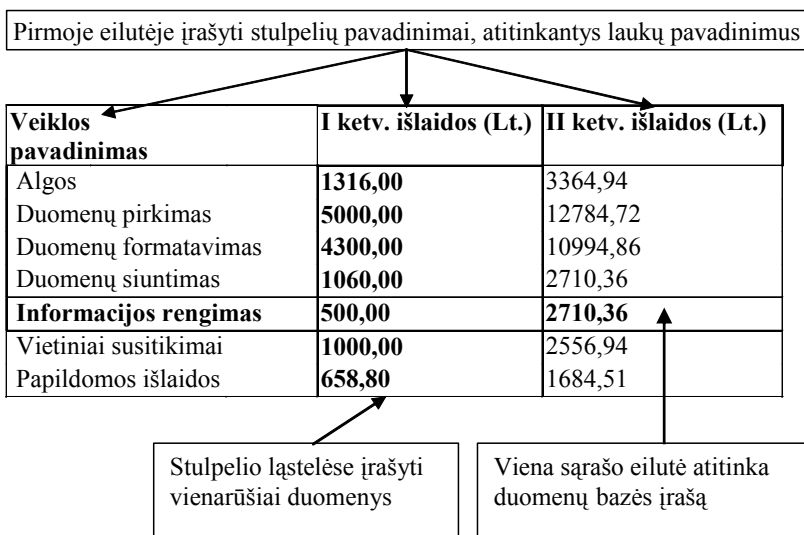
4.8. DUOMENŲ BAZIŲ IR SĄRAŠO ANALIZĖS KOMANDOS

Išmonių, organizacijų vadybos duomenys dažniausiai kaupiami ir apdorojami duomenų bazėse (*DB*). Duomenų bazėms aptarnauti sukurtos specialios programų sistemos, vadinamos duomenų bazių valdymo sistemomis (*DBVS*). Šiuo metu populiarios *DBVS* yra *Microsoft Access*, *ORACLE*, *My SQL*. Ankščiau buvo naudojamos *FoxPro Visual*, *DB2* ir kt. duomenų bazių valdymo sistemos.

Daugumoje šių *DBVS* duomenys tvarkomi pasitelkiant reliacinį duomenų modelį, grindžiamą matematine reliacinės algebros teorija.

Paprasčiausiai suprantamas reliacinis duomenų modelis – tai lentelėmis pateikta duomenų užrašymo forma, kurią sudaro eilutės (duomenų bazės įrašai), stulpeliai (duomenų bazės laukai).

Sąrašas (ang. *list*) – tai pavadinimus turinčių laukų, sutvarkytų eilutėmis, rinkinys. Sąrašą sudarančios lentelės laukuose saugomi vienerūšiai duomenys. Sąrašo komponentų samprata pateikiama paveikslėlyje.



4.35 pav. Sąrašą formuojantys komponentai

Iš pavyzdžio matyti sąrašą formuojančių komponentų struktūrą.

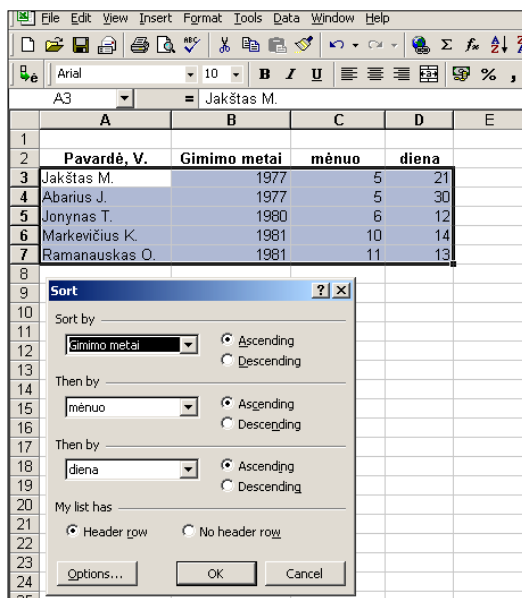
Pagrindiniai sąrašo bruožai:

- ✚ *MS Excel* automatiškai atpažįsta sąrašą, todėl galima bet kurioje lentelėje taikyti sąrašo analizės komandas.
- ✚ Sąrašo duomenys gali būti įvairiai rūšiuojami taikant *Data→Sort* komandą;

- ✚ Taikant *Data*→*Subtotals* komandą galima automatiškai rasti dalines sąrašo eilučių grupių sumas;
- ✚ Pasitelkiant *Data*→*Form* komandą labai paprasta peržiūrėti, surasti, pakeisti, pridėti arba išmesti sąrašo eilutes;
- ✚ Taikant *Data*→*Filter* komandą galima iš sąrašo išrinkti duomenis pagal nuolat kintančias vartotojų užklaudas.

4.8.1. Duomenų rikiavimas – patogi duomenų organizavimo priemonė

Excel programa pagal abėcėlę arba atvirkštine tvarka gali rikiuoti (išdėstyti, rūšiuoti) tekstinę ir skaitmeninę informaciją didėjimo arba mažėjimo tvarka.



4.36 pav. Duomenų rikiavimo komandos *Data*→*Sort* langas

Norint duomenis ląstelėse išdėstyti tam tikru būdu reikia pažymėti vientisą stačiakampę ląstelių sritį ir vykdyti meniu komandą *Data*→*Sort*; sąrašo *Sort by* išrinkti rūšiuojamą lauką. Tai gali būti

stulpelių numeriai, pavyzdžiui, *Column1*, jeigu lentelė neturi jų pavadinimų arba lentelės stulpelių pavadinimai įvesti vartotojo. Išsirinkus stulpelį reikia pažymėti vieną iš dviejų rūšiavimo parametru: abėcėlės tvarka (*Ascending*) arba atvirkštine negu abėcėlės tvarka (*Descending*). Aktyvus parametras *Header row* nurodo, kad lentelė turi antraštę. Duomenis galima rūšiuoti pagal tris skirtingus laukus.

Jeigu duomenys rūšiuojami daugiau negu pagal tris laukus, tai atliekama keletą kartų. Pavyzdžiui, jeigu duomenis reikia surūšiuoti pagal penkis laukus, iš pradžių rūšiuokite pagal du mažiausiai svarbius – ketvirtąjį ir penktąjį, antrą kartą rūšiuokite pagal pirmąjį, antrąjį ir trečiąjį laukus. Komandos *Sort* dialogo langelyje svarbesnis laukas įrašomas pirmiau.

4.8.2. Duomenų filtravimas

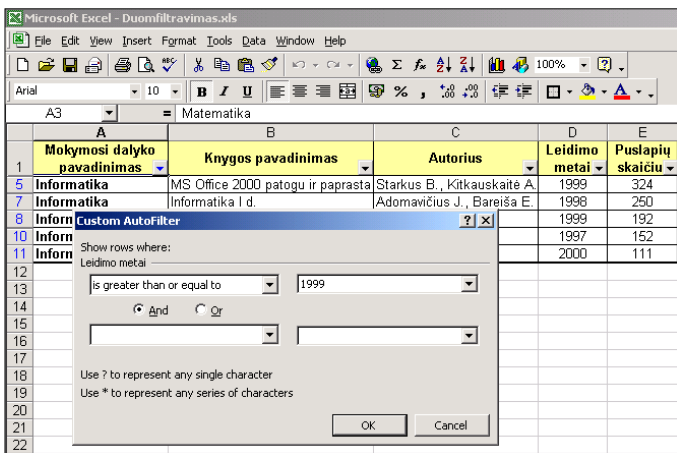
Atlikdamas lentelės duomenų filtravimą vartotojas gali greitai atrinkti duomenis pagal parametru reikšmes.

	A	B	C	D	E
	Mokymosi dalyko pavadinimas	Knygos pavadinimas	Autorius	Leidimo metai	Puslapių skaičius
1					
2	Istorija	Lietuvos istorija	Šapoka A.	1989	688
3	Matematika	Matematinės analizės pagrindai	Fichtengolcas G.	1987	451
4	Fizika	Bendroji fizika	Brazdžiūnas P.	1977	438
5	Informatika	MS Office 2000 patogu ir paprasta	Starkus B., Kitkauskaitė A.	1999	324
6	Istorija	Lietuvos istorija	Ivinskis Z.	1991	258
7	Informatika	Informatika I d.	Adomavičius J., Bareiša E.	1998	250
8	Informatika	Geltonieji interneto puslapiai	Valavičius E.	1999	192
9	Fizika	Fotonai ir branduoliai	Kitagorskis A.	1985	168
10	Informatika	Į pasaulį su internetu	Leonavičienė B.	1997	152
11	Informatika	Informatikos pradmenys IV dalis	Dagiienė V.	2000	111
12					

4.37 pav. Duomenų lentelė prieš atliekant filtravimą

Įvedus duomenis į lentelę vykdant filtravimo komandą *Data→Filter→AutoFilter* iš sąrašo galima išrinkti atitinkamus duomenis, pavyzdžiui, *Informatikos* literatūrą, išleistą ne anksčiau kaip iki 1999 metų. Leidimo metų sąrašę pasirinkę *Custom*, komandos dialogo lange *Custom AutoFilter* renkamės sąlygą *daugiau arba lygu*

(is greater than or equal to), nuroydamis konkrečią skaičiaus reikšmę, pavyzdžiui, 1999.



4.38 pav. Filtravimo sąlygų nustatymo langas

Atlikus filtravimą pagal anksčiau nurodytas sąlygas programa pateikia lentelę, kurioje įrašyti išrinkti duomenys.

	A	B	C	D	E
	Mokymosi dalyko pavadinimas	Knogos pavadinimas	Autorius	Leidimo metai	Puslapių skaičiu
5	Informatika	MS Office 2000 patogu ir paprasta	Starkus B., Kitkauskaitė A.	1999	324
8	Informatika	Getlonieji interneto puslapiai	Valavičius E.	1999	192
11	Informatika	Informatikos pradmenys IV dalis	Dagienė V.	2000	111

4.39 pav. Duomenų lentelė atlikus filtravimą

Norint atsisakyti filtravimo, reikia vykdyti tą pačią komandą – *Data*→*Filter*→*AutoFilter*.

4.9. KLAIDŲ REIKŠMĖS SKAIČIUOKLIŲ PROGRAMOJE

Klaidų atsiranda tada, kai neteisinga formulės sintaksė arba kai pagal parašytą formulę programa negali apskaičiuoti rezultato. Dažniausiai daromos klaidos: dalyba iš nulio, pažymėtoje ląstelėje neįrašytas skaičius, klaidingai sujungtos koordinatės, neteisingai nurodytos koordinatės ir pan. Kad būtų lengviau ištaisyti klaidas, reikia atsiminti programos pateikiamas pagrindinių daromų klaidų reikšmes ir jų paaiškinimus.

4.11 lentelė. Klaidų reikšmės

Klaidų reikšmės	Klaidos paaiškinimas
<i>#DIV/0!</i>	Negalima dalyba iš nulio. Reikšmę dalijate iš ląstelės, kurioje įrašytas nulis, arba ląstelė yra tuščia.
<i>#N/A</i>	Negalima reikšmė. Negalima apskaičiuoti funkcijos ar formulės rezultato, nes nėra reikiamos informacijos. Užrašydami funkciją pateikėte klaidingus argumentus.
<i>#NAME?</i>	Neatpažintas pavadinimas. Klaidingai parašėte funkcijos pavadinimą arba duomenų bloko koordinatės.
<i>#NULL!</i>	Klaidingai sujungtos koordinatės. Nurodant bendrus dviejų blokų adresus, jų koordinatės reikia sujungti tarpo ženklu.
<i>#NUM!</i>	Klaidingas skaičius. Funkcijos argumentuose įrašytas klaidingas skaičius.
<i>#REF!</i>	Klaidingos ląstelės koordinatės. Pašalinote ląsteles, nurodytas kitose formulėse.
<i>#VALUE!</i>	Klaidingas argumentas arba operandas. Matematinėje formulėje panaudojote ląstelę su tekstu ten, kur turi būti skaičius.

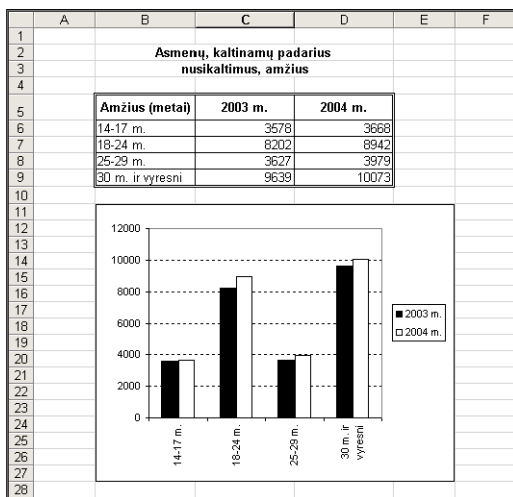
4.10. GRAFINIS DUOMENŲ PATEIKIMAS IR VIZUALIZAVIMAS

D iagramos yra viena vaizdžiausių duomenų pateikimo formų. Diagramos iliustruoja duomenų kitimo spartą bei lygina įvairius dydžius.

Excel programa pateikia šiuos diagramų tipus [24; 26]: stulpelinį (*Column*), juostinį (*Bar*), linijinį (*Line*), skritulinį (*Pie*), žiedinį (*Doughnut*), plotinį (*Area*), spindulinį (*Radar*), XY taškų (*Scatter*), paviršinį (*Surface*), kūginį (*Cone*), cilindrinį (*Cylinder*), piramidinį (*Pyramid*), akcijų (*Stock*), burbulinį (*Bubble*), kombinuotąjį (*Combination*).

Dažniausiai naudojamos stulpelinės, skritulinės ir linijinės diagramos. Šios diagramos gali būti pateikiamos dvimatėje (plokštuminės diagramos) ir trimatėje erdvėje (erdvinės diagramos).

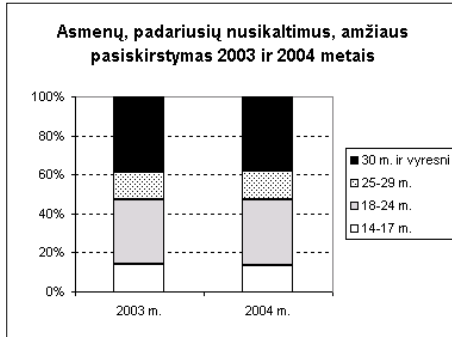
Stulpelių diagramomis patogu vaizduoti diskrečius duomenis. Stulpelių diagramose duomenys vaizduojami kaip vienodo pločio stačiakampiai. Tokios diagramos dar vadinamos histogramomis (šis žodis kilęs iš graikų kalbos žodžių *histos* – stulpas, *gramma* – užrašas). Šiose diagramose stulpelio aukštis proporcingas vaizduojamam dydžiui.



4.40 pav. Asmenų, kaltinamų padarius nusikaltimus 2003 ir 2004 metais, stulpelinė diagrama (pagal amžiaus grupes)

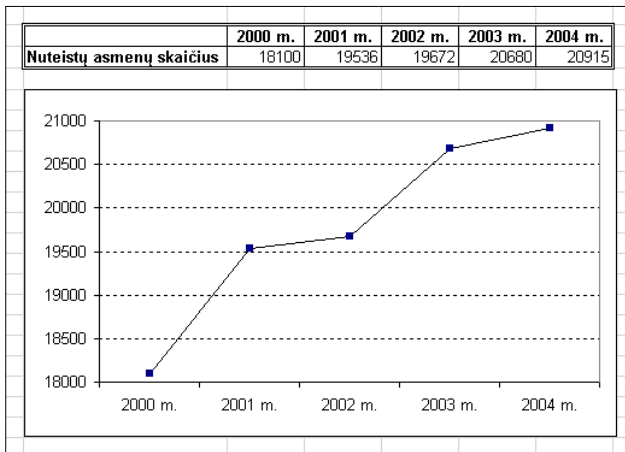
Diagramos, kuriose vaizduojami tie patys duomenys skirtingais laikotarpiais, vadinamos serijinėmis.

Stulpelinė santykinė diagrama naudojama, kai svarbus santykinis duomenų pasiskirstymas. Tokioje diagramoje stulpeliai atspindi vienaarūšių duomenų pasiskirstymą.



4.41 pav. Nusikaltimus padariusių asmenų amžiaus grupių pasiskirstymo 2003 ir 2004 metais stulpelinė santykinė diagrama

Norint pavaizduoti tolydžiai kintamų dydžių dinamiką naudojamos linijinės diagramos arba grafikai.



4.42 pav. 2000–2004 m. nuteistų asmenų skaičiaus dinamikos linijinė diagrama.

Skaičiuoklės *Excel* turi patogių diagramų sudarymo priemonių ir siūlo rinktis įvairių tipų diagramas – dvišakes ir trimates. Diagramos gali būti braižomos tame pačiame lape, šalia duomenų lentelių arba atskirame knygos lape.

Norint kokybiškai pateikti diagramą, reikia žinoti visus diagramos objektus, kurių formatus galima pasirinkti. Diagramos objektai yra šie:

- ✚ Diagramos sritis (*Chart Area*);
- ✚ Diagramos pavadinimas (*Chart Title*);
- ✚ Braižymo sritis (*Plot Area*);
- ✚ Legenda (*Legend*);
- ✚ Kategorijų ašis (*Category Axis*);
- ✚ Kategorijų ašies pavadinimas (*Category Axis Title*);
- ✚ Reikšmių ašis (*Value Axis*);
- ✚ Reikšmių ašių pavadinimas (*Value Axis Title*);
- ✚ Antroji reikšmių ašis (*Secondary Value Axis*);
- ✚ Antrosios reikšmių ašies pavadinimas (*Secondary Value Axis Title*);
- ✚ Pirmoji duomenų seka (*Series 1*);
- ✚ Antroji duomenų seka (*Series 2*) ir t.t.
- ✚ Duomenų žymės (*Data Labels*);
- ✚ Tinklelis (*Gridlines*);
- ✚ Duomenų lentelė (*Data Table*);
- ✚ Kritimo linija (*Drop Line*);
- ✚ Ribinė linija (*High-Low Line*);
- ✚ Pagrindas (*Floor*), esant trimatei diagramai;
- ✚ Sienuelė (*Wall*), esant trimatei diagramai;
- ✚ Kampas (*Corner*), esant trimatei diagramai.

Excel diagramos braižomos su diagramų vedliu *Chart Wizard*, kuris padeda atlikti darbą, pateikdamas keturis dialogo langelius (nurodydamas 4 žingsnius).

Norint nubrėžti diagramą pagal lentelėje esančius duomenis, reikia:

- ✚ Lentelėje pažymėti reikiamą duomenų sritį;
- ✚ Mygtuku *Chart Wizard* iš bendrosios mygtukų juostos iškviesti diagramos braižymo žymį;

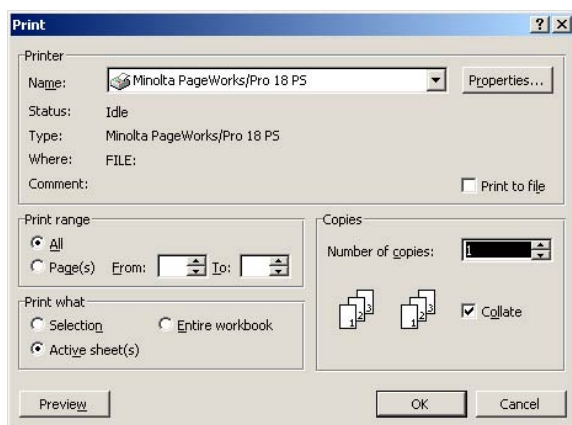
- Atliekant pirmąjį žingsnį (*Step 1 of 4*) pasirinkti diagramos tipą;
- Antrojo žingsnio metu (*Step 2 of 4*) pasirinkti duomenų išdėstymo būdą;
- Atliekant trečiąjį žingsnį (*Step 3 of 4*) pasirinkti diagramos apipavidalinimą;
- Atliekant ketvirtąjį žingsnį (*Step 4 of 4*) pasirinkti duomenų pateikimo vietą.

4.11. SKAIČIUOKLĖS DARBO KNYGOS SPAUSDINIMO BŪDAI

Skaičiuoklių *Excel* kuriamą knygą reikėtų ruošti tam spausdintuvui, kuriuo bus spausdinama. *Excel* knyga – tai dokumentas, galintis turėti daug lapų. Ją gali sudaryti ir diagramos, lentelės ir kita informacija. Ne visuomet reikia spausdinti visą dokumentą.

Spausdinant knygą dalimis, reikia vykdyti meniu komandą *File*→*Print*.

Iš šios komandos dialogo lango *Print* išplėstinio sąrašo *Name* išrenkamas spausdintuvo, kuriuo bus spausdinamas dokumentas, modelis.



4.43 pav. Spausdinimo komandos dialogo langas

Dalyje *Print range* pažymima, kuriuos lapus reikia spausdinti:

- ✚ *All* – spausdinti visus puslapius;
- ✚ *Pages* – spausdinti kai kuriuos puslapius, t. y. laukeliuose *From* (nuo), *To* (iki) įrašyti spausdinamų puslapių numerių intervalą.

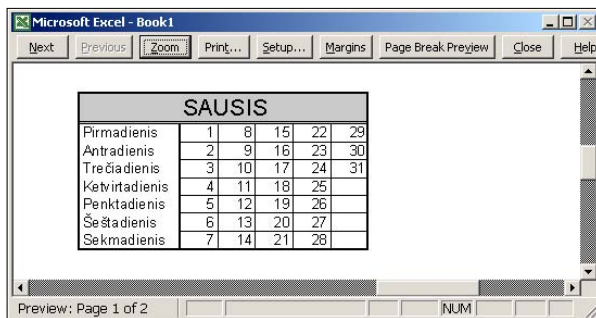
Dalyje *Print What* pažymima, ką reikia spausdinti:

- ✚ *Entire Workbook* – spausdinti visą knygą;
- ✚ *Active sheet(s)* – spausdinti aktyvaus lapo puslapius;
- ✚ *Selection* – spausdinti puslapyje pažymėtą sritį.

Dalyje *Copies* nurodomas kopijų skaičius.

Dokumento peržiūra prieš spausdinimą

Prieš spausdinant dokumentą naudinga pasižiūrėti, kaip jis atrodys išspausdintas. Peržiūros prieš spausdinimą režimas įjungiamas spaudžiant bendrosios paskirties įrankių juostos mygtuką *Print Preview* arba vykdant komandą *File*→*Print Preview*. Atliekant šį veiksma *Excel* programos lange bus matyti pirmasis spausdinamas lapas. *Print Preview* režimo metu pasikeičia ir ekrane matomi įrankiai.



4.44 pav. *File*→*Print Preview* komandos langas

Dirbant šiuo režimu galima atlikti šiuos veiksmus:

- ✚ Mygtukais *Next* ir *Previous* puslapis po puslapio peržiūrėti dokumentą;
- ✚ Mygtuku *Zoom* keisti puslapio vaizdo mastelį;
- ✚ Spaudžiant mygtuką *Setup* keisti spausdinamo lapo parametrus: matmenis, padėtį, paraštes ir paraščių užrašus;

- Spaudžiant mygtuką *Margins* keisti lapo paraštes;
- Spaudžiant mygtuką *Print* vykdyti spausdinimo komandą;
- Mygtuku *Close* peržiūrą baigti.

Spausdinamo lapo dydžio ir padėties parinkimas

Norint parinkti lapo dydį ir paraštes, reikia vykdyti meniu komandą *File*→*Page Setup* arba, esant dokumento peržiūros režimui (*Print Preview*), nuspausti mygtuką *Setup*.



4.45 pav. Spausdinamo puslapio parametrų nustatymo komandos langas

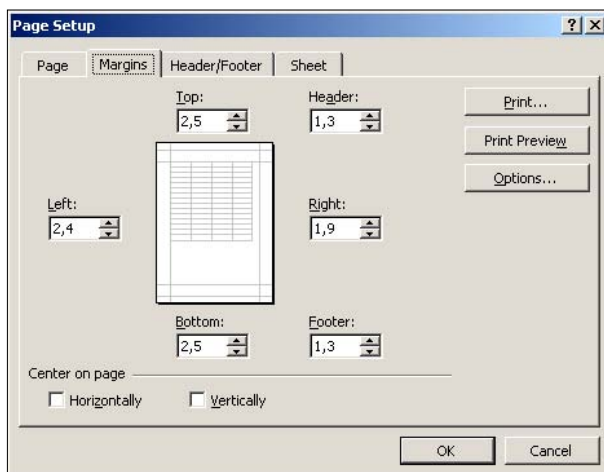
Page Setup dialogo lange skyrelyje *Page* dalyje *Orientation* galima nustatyti spausdinamo lapo padėtį: vertikalią (*Portrait*) arba horizontalią (*Landscape*).

Dalyje *Scaling* galima padidinti arba sumažinti spausdinamo objekto vaizdą. Pavyzdžiui: *Adjust to* – spausdinti 100 proc. dydžio vaizdą arba keisti spausdinamo vaizdo dydį, pavyzdžiui, 200 proc. vaizdą spausdinti dvigubai didesnę, 50 proc. – dvigubai mažesnę ir t. t. *Fit to* – spausdinamą vaizdą išdėstyti pagal ilgį ir plotį į tiek puslapių, kiek nurodoma dešinėje.

Dalyje *Paper Size* iš išplėstinio sąrašo galima išsirinkti lapo dydį, pavyzdžiui, A4 formato – 210x297 mm.

Dalyje *Print quality* nustatoma spausdinimo kokybė nurodant taškų skaičių colyje.

Skyrelyje *Margins* galima parinkti paraščių matmenis: *Top* – viršutinės paraštės, *Bottom* – apatinės paraštės, *Left* – kairiosios paraštės, *Right* – dešinėsios paraštės, *Header* nustatomas atstumas nuo viršutinio lapo krašto iki užrašo viršutinėje paraštėje. *Footer* nustatomas atstumas nuo apatinio lapo krašto iki užrašo apatinėje paraštėje.



4.46 pav. *File*→*Page Setup* komandos dialogo langas. Skyrelis *Margins* paraštėms parinkti

Skaičiuoklių *Excel* programa sukurtą lentelę galima išdėstyti spausdinamo lapo centre, horizontalia (*Horizontally*) arba vertikalio (*Vertically*) kryptimi.

4.12. SKAIČIUOKLIŲ PRAKTINIAI DARBAI IR METODINIAI NURODYMAI

4.12.1. Duomenų analizė skaičiuoklėse

1. Darbo tikslas

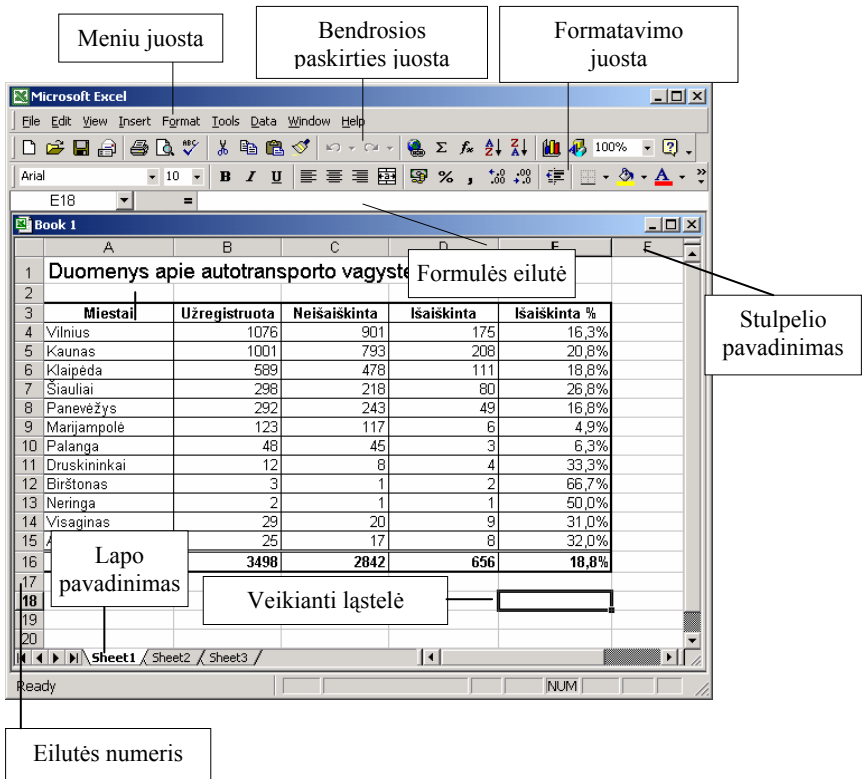
Susipažinti su skaičiuoklių darbo aplinka. Išmokti atlikti duomenų analizę, skaičiuoti procentus, įvertinant skirtinguose miestuose išaiškintų nusikaltimų skaičių.

2. Darbo užduotys

1. Iškvieskite *Microsoft Excel* programą. Susipažinkite su skaičiuoklės *Excel* pagrindiniu langu. Išmokite iškviešti nurodytas priemonių juostas – bendrosios paskirties, formatavimo, braižymo.
2. Parinkite A4 formato popieriaus lapo matmenis (210x297 mm). Parinkite paraščių matmenis: kairiosios – 1,9 cm, dešinėsios – 0,9 cm, viršutinės – 1,5 cm, apatinės – 1,5 cm.
3. Išmokite žymėti ląstelę, sritį, stulpelį, eilutę.
4. Įveskite duomenis apie autotransporto vagystes, pateiktus 1 lentelėje.
5. Atlikite skaičiavimus. Grafoje *Iš viso* suskaičiuokite, kiek yra užregistruotų, neišaiškintų ir išaiškintų autotransporto vagysčių. Sudarykite formulę, kuri suskaičiuotų išaiškintų vagysčių procentą nuo užregistruotų vagysčių skaičiaus.
6. Lentelę apipavidalinkite, suteikite jos antraštei norimą fono spalvą bei rėmelius.
7. Išsaugokite failą diskelyje (A:\), pavadinkite jį prasminiu pavadinimu, pavyzdžiui, *autovagyst.xls*.
8. Pabaikite dirbti su skaičiuokle *Excel*.

3. Metodiniai nurodymai

Susipažinkite su pagrindiniu skaičiuoklių *Excel* darbo langu.



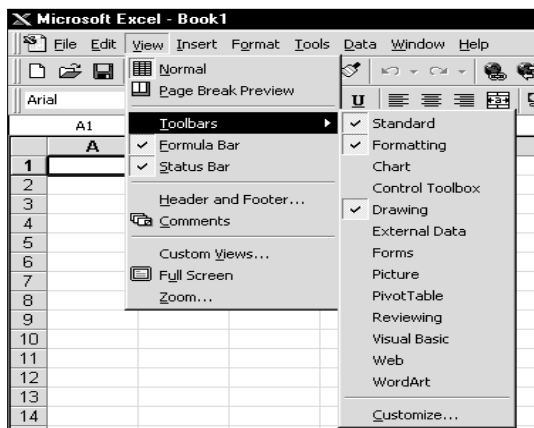
4.47 pav. Pagrindiniai *Excel* programos lango elementai

Kaip pasirinkti norimas mygtukų priemonių juostas?

Norėdami pasirinkti mygtukų juostas taip, kad jos būtų matyti *MS Excel* programos lange, vykdykite komandą *View*→*Toolbars*. Ties *Toolbars* pasirodžius sąrašui pažymėkite reikiamas priemones, pavyzdžiui, *Standard*, *Formatting*, *Drawing*.

Jeigu išvardytųjų mygtukų juostų nematyti, užduotį pakartokite.

Jeigu išvardytos priemonių juostos pažymėtos, paspauskite *View*, taip atšauksite komandas.

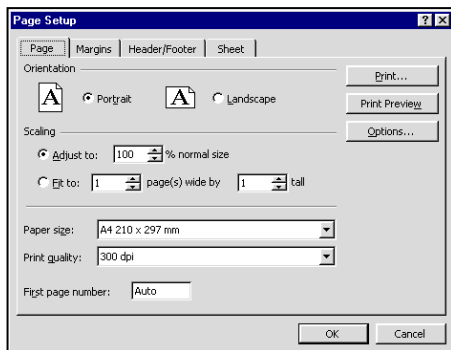


4.48 pav. Priemonių juostų įjungimas ir išjungimas vykdant pagrindinio meniu komandą *View*→*Toolbars*

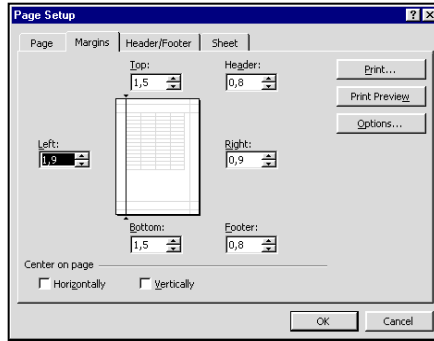
Kaip parinkti popieriaus lapo dydį, padėtį ir paraščių matmenis?

Pasirinkdami lapo formatą vykdykite komandą *File*→*Page Setup*. Skyrelyje *Page* parinkite A4 formato lapo matmenis (210x297 mm). Puslapio padėties nustatymo dalyje *Orientation* pažymėkite vertikalią lapo padėtį – *Portrait* (4.49 pav.).

Skyrelyje *Margins* nustatykite paraštes (4.50 pav.). Patikrinkite, ar parinkote norimus puslapio parametrus, pagrindiniame *Excel* lange sumažinkite mastelį, pavyzdžiui, 25 proc. Aktyviame lape (*Sheet*) bus matyti puslapių ribos. Iš naujo pasirinkite norimą mastelį.



4.49 pav. Popieriaus lapo formatavimas



4.50 pav. Popieriaus lapo paraščių nustatymas

Kaip išmokti žymėti ląstelę, sritį, stulpelį, eilutę?
Atlikite lentelėje nurodytus žymėjimo veiksmus.

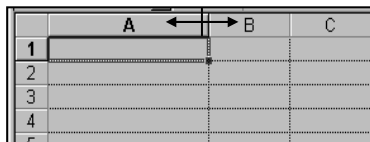
4.51 pav. Pažymėtų darbo lapo ląstelių ir blokų pavyzdžiai

4.12 lentelė. Ląstelių žymėjimas pele

Žymimas objektas	Veiksmo aprašymas
Ląstelė	Paspausti kairijį pelės klavišą ties išskiriama ląstele
Sritis	Paspausti kairijį pelės klavišą ties kampine srities ląstele ir traukti žymyklę iki kito srities kampo
Eilutė (stulpelis)	Paspausti kairijį pelės klavišą ties eilutės numeriu (stulpelio pavadinimu)
Sudėtinė sritis	Pažymėti pirmąją sritį, tada laikant nuspaudus klavišą <Ctrl>, pažymėti papildomas sritis
Visa lentelė	Nuspausti kairijį pelės klavišą ties stačiakampiu, esančiu stulpelių ir eilučių pavadinimų sankirtoje – virš 1, i kairę nuo A.

Kaip pakeisti stulpelio plotį?

Į skaičiuoklės darbo lapą įveskite duomenis, pateiktus 4.13 lentelėje. Jeigu lentelės stulpelio plotį reikia padidinti (sumažinti), tai atlikite traukdami pelę, kaip parodyta paveikslėlyje.



4.52 pav. Stulpelio pločio keitimas pele

Kaip į lentelę įvesti duomenis?

Lentelės pavadinimą pradėkite rašyti nuo kairiojo lapo krašto. Du kartus spausdami kairįjį pelės klavišą pažymėkite ląstelę A1. 4.47 paveikslėlyje matyti bendras lentelės vaizdas. Spausdami kairįjį pelės klavišą pažymėkite ląstelę, į kurią įvesite duomenis, ir juos surinkite. Duomenis į lentelę įveskite judėdami nuo ląstelės prie ląstelės rodyklėmis (klaviatūroje) arba pele.

4.13 lentelė. Duomenys apie autotransporto vagystes

	A	B	C	D	E
1	Duomenys apie autotransporto vagystes				
2					
3	Miestai	Užregistruota	Neišaiškinta	Išaiškinta	Išaiškinta %
4	Vilnius	1076	901		
5	Kaunas	1001	793		
6	Klaipėda	589	478		
7	Šiauliai	298	218		
8	Panevėžys	292	243		
9	Marijampolė	123	117		
10	Palanga	48	45		
11	Druskininkai	12	8		
12	Birštonas	3	1		
13	Neringa	2	1		
14	Visaginas	29	20		
15	Alytus	25	17		
16	Iš viso				

Kaip atlikti skaičiavimus lentelėje?

Pagrindiniai aritmetiniai veiksmai: + (sudėtis), – (atimtis), * (daugyba), / (dalyba), ^ (kėlimas laipsniu), % (procentai).

Pažymėkite ląstelę *Iš viso–Užregistruota*, t. y. B16 ir paspauskite mygtuką *AutoSum* Σ . Sumuojami duomenys taps aktyvūs. Antrą kartą paspaudus mygtuką *AutoSum* ląstelėje pasirodys rezultatas. Šiuo veiksmu susipažinome su dažniausiai naudojama sumavimo funkcija =SUM(B4:B15). Analogiškai atlikite skaičiavimą ląstelėje *Iš viso–Neišaiškinta* =SUM(C4:C15) (ląstelėje C16 galite parašyti funkciją =SUM(C4:C15) ir nuspauskite klavišą <Enter>.

	A	B	C	D	E
1	Duomenys apie autotransporto vagystes				
2					
3	Miestai	Užregistruota	Neišaiškinta	Išaiškinta	Išaiškinta %
4	Vilnius	1076	901	=B4-C4	
5	Kaunas	1001	793		
6	Klaipėda	589	478		
7	Šiauliai	298	218		
8	Panevėžys	292	243		
9	Marijampolė	123	117		
10	Palanga	48	45		
11	Druskininkai	12	8		
12	Birštonas	3	1		
13	Neringa	2	1		
14	Visaginas	29	20		
15	Alytus	25	17		
16	<i>Iš viso</i>	3498	2842		

Parašę formulę spauskite klavišą <Enter>

4.53 pav. Išaiškintų autotransporto vagysčių skaičiavimas sudarant formulę

Suaktyvinkite ląstelę D4 (ties antrašte *Išaiškinta*) ir įrašykite formulę =B4–C4, pagal kurią bus skaičiuojamas išaiškintų vagysčių skaičius.

Kaip kopijuoti formulę?


Įvestąją formulę kopijuokite į ląstelių sritį, vykdydami komandą *Edit*→*Fill*→*Down* arba traukdami žemyn pelės žymiklį.

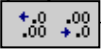
Book 1					
	A	B	C	D	E
1	Duomenys apie autotransporto vagystes				
2					
3	Miestai	Užregistruota	Neišaiškinta	Išaiškinta	Išaiškinta %
4	Vilnius	1076	901	175	
5	Kaunas	1001	793		
6	Klaipėda	589	478		
7	Šiauliai	298	218		
8	Panevėžys	292	243		
9	Marijampolė	123	117		
10	Palanga	48	45		
11	Druskininkai	12	8		
12	Birštonas	3	1		
13	Neringa	2	1		
14	Visaginas	29	20		
15	Alytus	25	17		
16	<i>Iš viso</i>	3498	2842		

Įvestąją formulę kopijuokite traukdami žemyn paveikslėlyje parodytą pelės žymiklį.

4.54 pav. Formulės kopijavimas pele

Kaip skaičiuoti procentus?

Ląstelėje E4 ties antrašte *Išaiškinta %* parašykite formulę $=D4/B4$, paspauskite klavišą *<Enter>* ir gausite rezultatą dešimtaine trupmena. Norint gauti procentinį rezultato pavidalą formatavimo (*Formatting*) priemonių juostoje spaudžiamas mygtukas *Percent Style* .

Ženklos po kablelio galima suvienodinti pažymint ląsteles, kuriose tai reikia atlikti, ir paspaudžiant formatavimo (*Formatting*) įrankių juostos mygtukus  jie didina ir mažina ženklų skaičių po kablelio.

Kaip apipavidalinti lentelę?

Norėdami įrėminti lentelę, ją pažymėkite ir iš mygtuko *Borders* išplėstinio sąrašo parinkite patikusį rėmelių variantą.

Rėmeliai Spalvinimas Šrifto spalva
Border *Fill Color* *Font Color*



4.55 pav. Formatavimo priemonių juosta



Jeigu norite nuspalvinti lentelės ląsteles, pavyzdžiui, lentelės antraštę, ją pažymėkite ir mygtuko *Fill Color* spalvinimo skydelyje parinkite spalvą. Šiuos skydelius galima perkelti į kitą vietą (*tear off*) – perkelkite skydelį į darbo lapą galėsite jį daug kartų naudoti ląstelių formatams tvarkyti.

Pažymėkite tekste ląsteles ir pakeiskite jose esančių duomenų šrifto spalvą. Tai atliekama iš mygtuko *Font Color* išplėstinio sąrašo pasirenkant šrifto spalvą.

Įvairesni įrėminimo būdai parenkami vykdant komandą *Format* → *Cells* → *Border*.

Kaip failą išsaugoti?



Norint išsaugoti skaičiuoklių darbo knygą reikia vykdyti komandą *File* → *Save As*. Komandos dialogo lange reikia:

-  Išplėstiniame sąraše *Save in* nurodyti aplanką, kuriame bus saugomas dokumentas;
-  Teksto lauke *File Name* įrašyti pavadinimą, kuriuo bus saugomas failas, pavyzdžiui, *autovagyst* (.xls plėtinio galima nerašyti). Atlikite šiuos veiksmus ir paspauskite mygtuką *Save*.

Programos *Excel* lango pavadinimų juostoje turi atsirasti failo pavadinimas (pvz., *autovagyst.xls*). Jeigu failo pavadinimo nėra, užduotį pakartokite.

Kaip įkelti (atverti) diske esantį failą?

Norint įkelti iš disko dokumentą, reikia vykdyti komandą *File* → *Open*. Komandos dialogo lange *Open* reikia:

-  Išplėstiniame sąraše *Look in* nurodyti aplanką, kuriame yra saugomas dokumentas;
-  Sąraše pažymėti failo pavadinimą ir paspausti mygtuką *Open*.

Kontroliniai klausimai

1. Kokius žinote formulės sudarymo būdus?
2. Kaip galima kopijuoti formulę?
3. Kaip pataisyti ląstelės turinį?
4. Kokias žinote priemones ląstelių formatams tvarkyti?

4.12.2. Duomenų vizualizavimas. Diagramų sudarymas

1. Darbo tikslas

Susipažinti su duomenų vizualizavimo skaičiuoklėse pagrindais. Išmokyti grafiškai pateikti duomenis, taikant skirtingus diagramų tipus.

2. Darbo užduotys

1. Iškvieskite *Microsoft Excel* programą.
Nustatykite A4 formato popieriaus lapo matmenis (210x297 mm).
Parinkite paraščių matmenis: kairiosios – 1,9 cm, dešinėsios – 0,9 cm, viršutinės – 1,5 cm, apatinės – 1,5 cm.
2. Įveskite lentelėje pateiktus Baltijos šalių nusikalstamumo rodiklių duomenis ir apskaičiuokite vidurkį. Įrėminkite lentelę, jos antraštei suteikite norimą fono spalvą.
3. Tame pačiame darbo lape nubraižykite dvimatę stulpelinę diagramą „Nusikaltimų dinamika Baltijos valstybėse 2000–2004 metais“.
4. Tame pačiame darbo lape nubraižykite dvimatę skritulinę diagramą „Nusikaltimų dinamika Baltijos valstybėse 2000–2004 metais“.
5. Tame pačiame darbo lape nubraižykite dvimatę linijinę diagramą „Nusikaltimų dinamika Baltijos valstybėse 2000–2004 metais“.
6. Nukopijuokite diagramą ir ją pakeiskite kito tipo diagrama. Parinkite jums patinkančią ir tinkančią vaizduoti pateiktuosius duomenis diagramą.

3. Metodiniai nurodymai

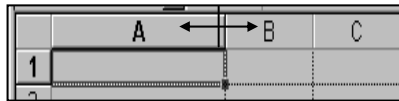
Kaip taisyklingai įvesti duomenis, užrašyti lentelės pavadinimą ir suskaičiuoti vidurkį?

Įveskite lentelėje pateiktus duomenis. Suskaičiuokite vidurkį.

4.14 lentelė. Nusikalstamumo dinamika Baltijos valstybėse
2000-2004 metais (nusikalstamumas 10 000 gyventojų,
visų tarnybų duomenimis)

Valstybė	2000 m.	2001 m.	2002 m.	2003 m.	2004 m.	Vidurkis
Lietuva	210,90	208,30	222,50	214,50	210,00	
Latvija	147,90	180,20	211,30	215,00	219,00	
Estija	314,40	356,70	419,80	428,00	435,00	

Lentelės pavadinimą pradėkite rašyti nuo kairiojo lapo krašto. Įveskite 4.14 lentelėje pateiktus duomenis. Jeigu lentelės stulpelio plotį reikia padidinti (sumažinti), tai atlikite traukdami pele, kaip parodyta paveikslėlyje.



4.56 pav. Stulpelio pločio keitimas pele

Įvedus duomenis, galima atlikti analizės veiksmus.

Vidurkį galima skaičiuoti pasitelkiant vidurkio funkciją $=AVERAGE(B5:F5)$. Įterpti funkciją galima vykdant komandą *Insert*→*Function* ir iš sąrašo pasirenkant reikiamą funkciją.

Valstybė	2000 m.	2001 m.	2002 m.	2003 m.	2004 m.	Vidurkis
Lietuva	210,90	208,30	222,50	214,50	210,00	213,24
Latvija	147,90	180,20	211,30	215,00	219,00	
Estija	314,40	356,70	419,80	428,00	435,00	

4.57 pav. Aritmetinio vidurkio skaičiavimas pagal formulę

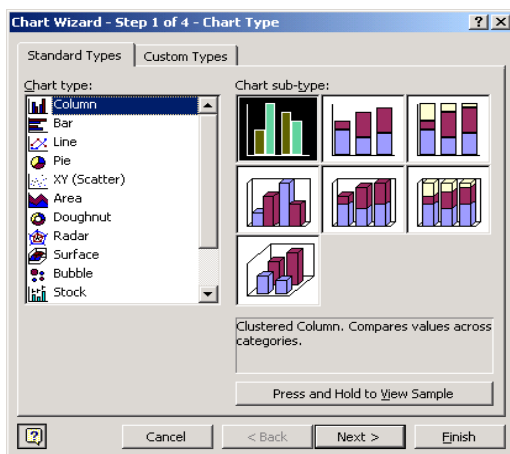
Kaip vaizduoti duomenis dvimate stulpeline diagrama?

Lentelėje pažymėkite duomenis, kuriuos norite vaizduoti dvimate stulpeline diagrama (pvz., 4.60 pav. pavaizduoti trijų Baltijos valstybių 2000–2004 m. laikotarpio duomenys) ir iškvieskite bendrojoje mygtukų juostoje esantį diagramų vedlį (*Chart Wizard*). Jis pateikia keletą dialogo langelių, kuriuose vartotojas paeiliui pasirinkdamas pateiktus objektus ir parametrus nubraižo diagramą.

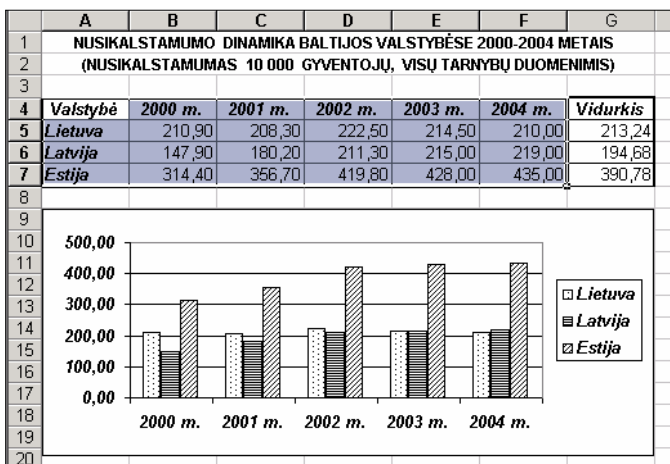


4.58 pav. Bendrosios paskirties priemonių juosta ir mygtukas *Chart Wizard* diagramų vedliui iškviesti

Dialogo lange *Chart Wizard – Step 1 of 4* pažymėkite dvimatės stulpelinės diagramos tipą. Diagramų vedliu paeiliui (*Next*) pasirinkite reikiamus parametrus ir programa automatiškai sudarys diagramą.



4.59 pav. Pirmasis diagramų vedlio dialogo langelis

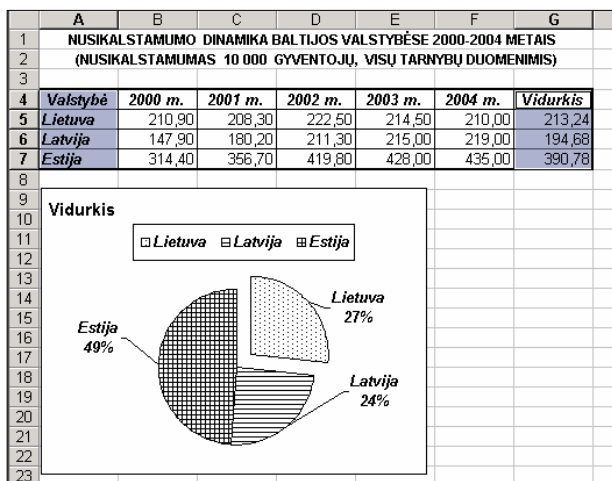


4.60 pav. Dvimatė stulpelinė serijinė diagrama

Kaip nubraižyti skritulinę diagramą?

Atlikite lentelėje nurodytus žymėjimo veiksmus, t. y. pažymėkite valstybes ir jų nusikalstamumo vidurkius, iškvieskite diagramų vedlį *Chart Wizard*.

Užduotį atlikite analogiškai kaip ir braižydami stulpelinę diagramą, tik pirmojo veiksmo metu pasirinkite skritulinės diagramos tipą – *Pie*.

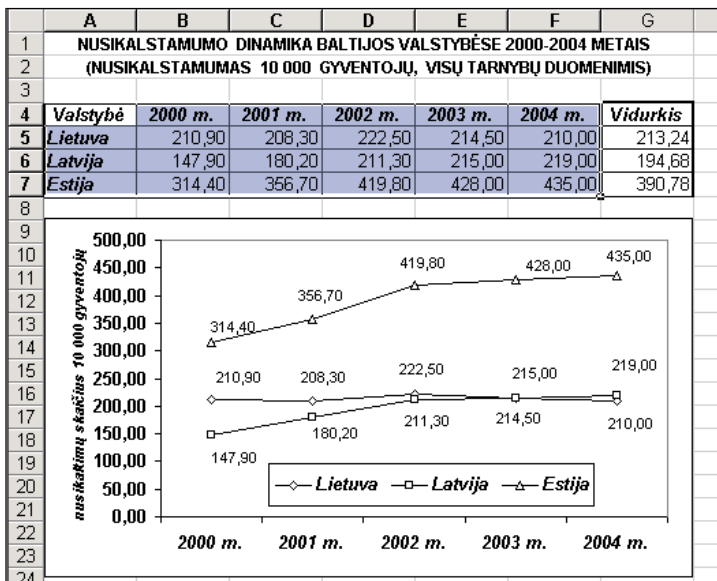


4.61 pav. Dvimatė skritulinė diagrama pateikiant aritmetinio vidurkio duomenis procentais

Kaip nubraižyti linijinę diagramą?

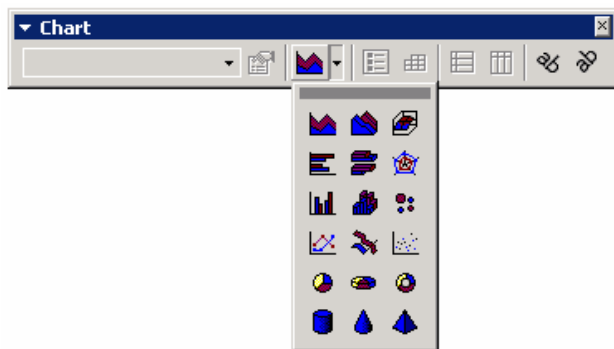
Lentelėje pažymėkite reikiamus duomenis, iškvieskite diagramų vedlį *Chart Wizard*.

Pažymėkite linijinę diagramą ir pasirinkite reikiamus parametrus diagramai pateikti.



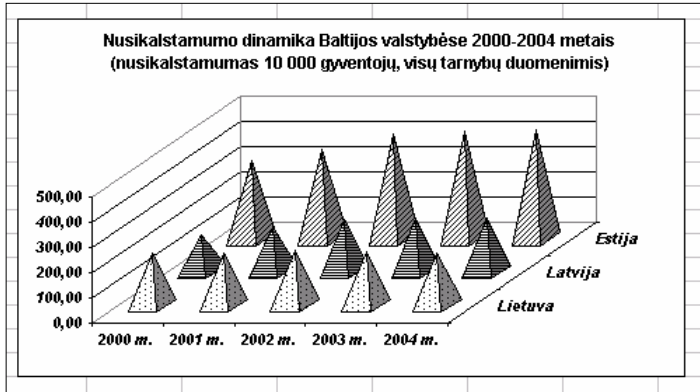
4.62 pav. Dvimatė linijinė diagrama, vaizduojanti nusikaltimų dinamiką nuo 2000 iki 2004 metų

Kaip vieno tipo diagramą keisti kita?



4.63 pav. Diagramų priemonių juostos mygtukai

Darbo lange nesant diagramų keitimo priemonių (*Chart*), vykdykite komandą *View*→*Toolbars* ir jas iškvieskite. Nukopijuokite vieną iš lape jau esančių diagramų, jos kopiją pažymėkite ir mygtuko *Chart Type* išplėstiniame sąraše parinkite tinkančią vaizuoti reikiamus duomenis diagramą. Susipažinkite su *Chart* juostos mygtukais.



4.64 pav. Trimatė piramidinė diagrama, turinti tris ašis: reikšmių ašį (Z) – aukščiui, kategorijų ašį (X) – pločiui ir sekų ašį (Y) – storiui (gyliui) vaizduoti

Kontroliniai klausimai

1. Kokius žinote diagramų sudarymo būdus?
2. Kaip nustatomos žymių reikšmės linijinėje diagramoje?
3. Kaip parinkti parametrus, kad skritulinėje diagramoje būtų vaizduojami procentai?
4. Kaip vieno tipo diagramą pakeisti kita?
5. Kaip suteikti duomenų vaizdams diagramoje (stulpeliams, skritulio skiltims, linijoms) kitas spalvas, raštus.
6. Kaip įrašyti diagramos ir ašių pavadinimus?
7. Diagramos teksto redagavimo priemonės ir būdai.

4.12.3. Pajamų planavimas ir analizė

1. Darbo tikslas

Susipažinti su funkcijų taikymo pagrindais planuojant bei vertinant pajamas.

2. Darbo užduotys

1. Iškviestite *Microsoft Excel* programą. Susipažinkite su skaičiuoklės *Excel* funkcijomis. Išmokite iškviesti metodiniuose nurodymuose pateiktas funkcijas, išsiaiškinkite jų paskirtį.

2. Parinkite A4 formato popieriaus lapo matmenis (210x297 mm).

Parinkite sudaromos lentelės paraščių matmenis, pavyzdžiui, kairiosios – 1,9 cm, dešinėsios – 0,9 cm, viršutinės – 1,5 cm, apatinės – 1,5 cm.

3. Sudarykite darbo lapą planuojamoms I, II ir III ketvirčių biudžeto pajamoms apskaičiuoti.

I ketvirčio pajamos pateiktos 4.15 lentelėje. Kitų ketvirčių planuojamas pajamas skaičiuokite remdamiesi I ketvirčio duomenimis.

II ketvirčio planuojamas pajamas apskaičiuokite pagal toliau pateiktas sąlygas. Įvertinkite I ketvirčio biudžeto pajamų dydį. Jeigu jos yra mažesnės negu 600000 tūkst. Lt, skaičiuodami II ketvirčio pajamas jas padidinkite 3 proc. Jeigu I ketvirčio pajamos šios sąlygos netenkina (t. y. jos yra didesnės arba lygios 600000 tūkst.), skaičiuodami II ketvirčio pajamas joms suteikite tokią pačią reikšmę kaip ir I ketvirtyje, t. y. jų nekeiskite.

Skaičiavimams atlikti naudokite loginę funkciją *IF*.

Planuojamas III ketvirčio pajamas apskaičiuokite pagal šias sąlygas. Skaičiuodami III ketvirčio pajamas, įvertinkite I ketvirčio biudžeto pajamų dydį. Jeigu jos viršija 1000000 tūkst. Lt, skaičiuodami III ketvirčio pajamas jas padidinkite 3 proc., sumas, kurios viršija 600000 tūkst. Lt ir yra mažesnės arba lygios 1000000 tūkst. Lt, padidinkite 2 proc., sumas, mažesnes arba lygias 600000 tūkst. Lt, padidinkite 1 proc.

Skaičiavimams atlikti naudokite dviejų lygmenų loginę funkciją *IF*.

4. Pajamų vidurkį suskaičiuokite pasitelkdami funkciją *AVERAGE*, „iš viso“ – naudodami funkciją *SUM*. Naudodami *MAX* ir *MIN* funkcijas pateikite maksimalias ir minimalias pajamas.
5. Pateikite minimalias ir maksimalias sumas, ties sumomis įterpkite jas aiškinantį komentarą.
6. Lentelę apipavidalinkite: parinkite rėmelius, antraštei suteikite tamsesnę foną.
7. Failą įrašykite į diską, suteikite failui pavadinimą, kuris atspindėtų jame saugomos informacijos prasmę, pavyzdžiui, *pajamos.xls*.

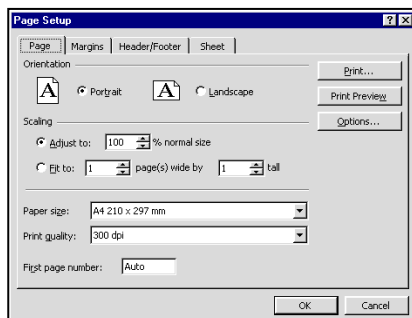
4.15 lentelė. Duomenys apie planuojamas biudžeto pajamas

B i u d ž e t o p a j a m o s (t ū k s t . L t)				
Pajamų rūšys	I ketvirtis	II ketvirtis	III ketvirtis	Vidurkis
Fizinių asmenų pajamų mokestis	782696			
Juridinių asmenų pelno mokestis	289183			
Pridėtinės vertės mokestis	1266293			
Akcizai	498025			
Tarptautinės prekybos ir sandėrių mokesčiai	251647			
Kitos mokestinės pajamos	205300			
Nemokestinės pajamos	106459			
<i>Iš viso</i>				
<i>Minimalios pajamos</i>				
<i>Maksimalios pajamos</i>				

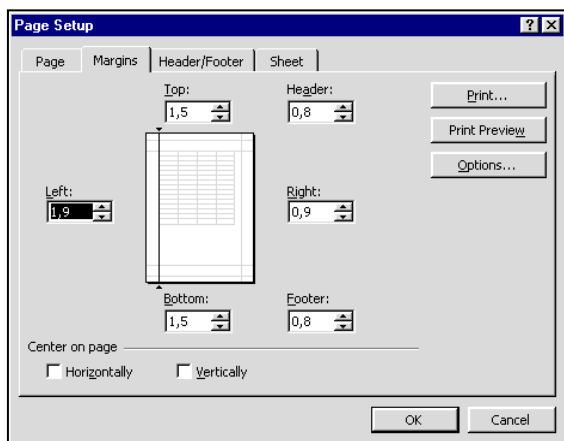
3. Metodiniai nurodymai

Kaip parinkti darbo lapo paraščių matmenis ir lapo dydį?

Vykdykite komandą *File*→*Page Setup*. Skyrelyje *Page* parinkite lapo dydį – A4, vertikalią lapo padėtį – *Portrait*. Paraščių skyrelyje *Margins* nustatykite paraštes. Patikrinkite, ar parinkote norimus puslapio parametrus. Pagrindiniame *Excel* lange mastelį sumažinkite iki 25 proc., aktyviame lape matysite puslapių ribas, tada mastelį vėl padidinkite iki 100 proc. Darbo lango dešinėje bus matyti punktyrinė linija – jūsų lapo dešinioji riba.



4.65 pav. Komandos *File*→*PageSetup* dialogo langas lapo dydžiui ir padėčiai parinkti



4.66 pav. Komandos *File*→*Page Setup* dialogo langas paraštėms parinkti

Kaip duomenis taisyklingai įvesti į lentelę?

Į lentelę įveskite reikiamus duomenis. Jeigu lentelės stulpelio plotį reikia padidinti (sumažinti), tai atlikite.

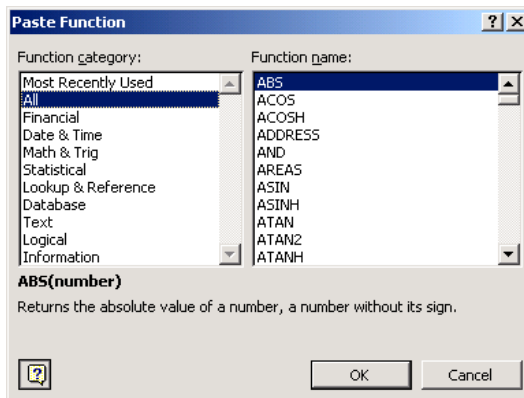
Lentelės pavadinimą pradėkite rašyti nuo kairiojo lapo krašto, ląstelėje A1. Bendras lentelės vaizdas matyti toliau esančiame *Excel* lange (4.68 pav.). Spausdami kairįjį pelės klavišą pažymėkite ląstelę, į kurią įvesite duomenis, ir juos įveskite. Įvesdami duomenis į lentelę iš vienos ląstelės į kitą judėkite rodyklėmis (*klaviatūros*) arba pele.

Funkcijų taikymas sudarant formules

Funkcijos taikomos įvairiems skaičiavimams palengvinti, jas galima derinti tarpusavyje bei naudoti sudėtingesnėms formulėms sudaryti. Kiekviena funkcija turi savo sintaksę. Tinkamai įrašius argumentus, funkcija apskaičiuoja rezultatą ir įrašo jį į ląstelę, kurioje ji yra sudaroma.

Funkcijų grupės yra tokios:

- ✚ Visos funkcijos (*All*);
- ✚ Finansinės (*Financial*);
- ✚ Datos ir laiko (*Date&Time*);
- ✚ Matematinės ir trigonometrinės (*Math&Trig*);
- ✚ Statistinės (*Statistical*);
- ✚ Paieškos, nuorodų (*Lookup&Reference*);
- ✚ Duomenų bazės (*Database*);
- ✚ Tekstinės (*Text*);
- ✚ Loginės (*Logical*);
- ✚ Informacinės (*Information*).



4.67 pav. *Paste Function* dialogo langas

Funkcijos gali būti įterpiamos į formulę įvedimo eilutėje įrašant jų pavadinimą ir skliaustuose nurodant argumentų sąrašą arba iškviečiamos automatiškai vykdant komandą *Insert*→*Function*.

Atsivėrusiame lange *Paste Function* galima pasirinkti funkcijos tipą (*Function category*) ir nurodyti funkcijos pavadinimą (*Function name*).

žesnės negu 600000 tūkst. Tada II ketvirtyje jas didinsime 3 proc., priešingu atveju jos liks tokios pačios kaip ir I ketvirtyje.

II ketvirčio pajamoms pateikti pagal anksčiau nurodytas sąlygas ląstelėje C3 sudaroma formulė

$$=IF(B3<600000; B3+B3*3%; B3).$$

Įvestą formulę kopijuojame taip: gauname II ketvirčio planuojamas pajamas pagal pateiktas šios užduoties sąlygas.

The screenshot shows an Excel spreadsheet titled "Microsoft Excel - Pajamos.xls". The formula bar shows the formula for cell C3: $=IF(B3<600000; B3+B3*3%; B3)$. The spreadsheet contains the following data:

	A	B	C	D	E
1	Biudžeto pajamos (tūkst. Lt)				
2	Pajamų rūšys	I ketvirtis	II ketvirtis	III ketvirtis	Vidurkis
3	Fizinių asmenų pajamų mokestis	782696	782696		
4	Juridinių asmenų pelno mokestis	289183			
5	Pridėtinės vertės mokestis	1266293			
6	Akcizai	498025			
7	Tarptautinės prekybos ir sandėrių mokesčiai	251647			
8	Kitos mokestinės pajamos	205300			
9	Nemokestinės pajamos				
10	<i>iš viso</i>				
11	<i>Minimalios pajamos</i>				
12	<i>Maksimalios pajamos</i>				

A callout box is overlaid on the spreadsheet with the text: "Įveskite funkciją, suskaičiuokite rezultata ir kopijuokite funkciją lentelėje."

4.68 pav. II ketvirčio planuojamų pajamų skaičiavimas naudojant *IF* funkciją

Sudarant formulę III ketvirčio pajamoms skaičiuoti *IF* funkcija įterpiama į argumentą, nes sąlyga yra sudėtingesnė.

III ketvirčio rezultatams gauti ląstelėje D3 sudaroma formulė


$$=IF(B3>1000000; B3+B3*3%; IF(B3>600000; B3+B3*2%; B3+B3*1%))$$

arba, užrašant trumpiau,

$$=IF(B3>1000000; B3*1,03; IF(B3>600000; B3*1,02; B3*1,01))$$

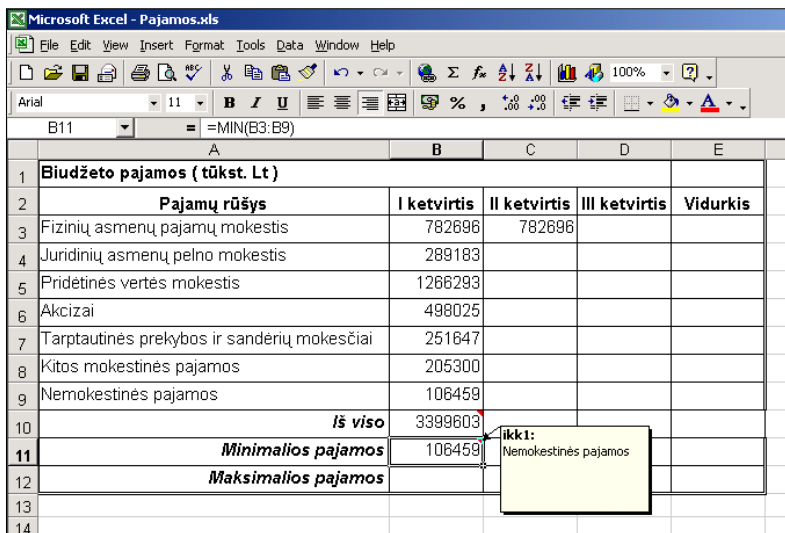
Nukopijavę įvestąją formulę gauname III ketvirčio visų planuojamų rūšių pajamas.

Kaip apskaičiuoti ketvirčių duomenų sumas?

Eilutėje *Iš viso* išskirkite vietą rezultatui – ląstelę B10 ir spauskite mygtuką *AutoSum*  – sumuojami duomenys taps aktyvūs (jeigu aktyvūs duomenys jūsų netenkina, galite pažymėti juos pele).

Kaip įterpti komentarą?

Pasirinkite ląstelę, į kurią norite įterpti komentarą, ir vykdykite komandą *Insert* → *Comment*.



The screenshot shows a Microsoft Excel spreadsheet titled "Pajamos.xls". The active cell is B11, containing the formula "=MIN(B3:B9)". A comment box is attached to cell B11, containing the text "Ikk1: Nemokestinės pajamos". The table data is as follows:

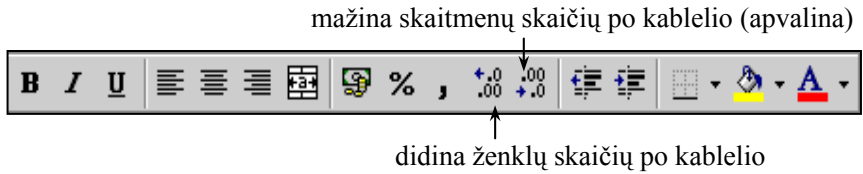
	A	B	C	D	E
1	Biudžeto pajamos (tūkst. Lt)				
2	Pajamų rūšys	I ketvirtis	II ketvirtis	III ketvirtis	Vidurkis
3	Fizinių asmenų pajamų mokestis	782696	782696		
4	Juridinių asmenų pelno mokestis	289183			
5	Pridėtinės vertės mokestis	1266293			
6	Akcizai	498025			
7	Tarptautinės prekybos ir sandėrių mokesčiai	251847			
8	Kitos mokestinės pajamos	205300			
9	Nemokestinės pajamos	106459			
10	Iš viso	3399603			
11	Minimalios pajamos	106459			
12	Maksimalios pajamos				
13					
14					

4.69 pav. Ląstelės komentario pavyzdys

Norint, kad komentarai būtų matyti tik ląstelei esant aktyviai, reikia vykdyti komandą *Tools* → *Options* → *View* ir įjungti *Comment Indicator Only*.

Kaip suvienodinti skaitmenų skaičių po kablelio?

Skaitmenų skaičių po kablelio galima suvienodinti pažymint ląsteles, kuriose tai reikia atlikti, ir spaudžiant formatavimo (*Formatting*) įrankių juostos mygtukus, kurių vienas mažina, o kitas didina ženklų skaičių po kablelio (4.70 pav.).



4.70 pav. Formatavimo priemonių juosta

Kaip apipavidalinti lentelę?

Lentelei apipavidalinti naudokite formatavimo priemonių juostos mygtukus. Norėdami įrėminti lentelę pažymėkite ją ir iš mygtuko *Borders* išplėstinio sąrašo parinkite patinkantį rėmelio variantą.

Foną suteiksite pažymėdami lentelės antraštę ir mygtuko *Fill Color* išplėstiname sąrašo parinkdami spalvą.

Pažymėkite tekste atskiras ląsteles, pakeiskite jose šrifto spalvą. Šrifto spalvą parinkite mygtuko *Font Color* išplėstiname sąrašo. Pažymėkite ląsteles, kurias norite sujungti, pavyzdžiui, rašydami pavadinimą (A1:D1) *Biudžeto pajamos* ir paspauskite mygtuką *Merge and Center*.

Įvairesnių rėmelio variantų galima atrasti vykdant komandą *Format*→*Cells*→*Border*. Tuomet komandos dialogo lange galima išsirinkti rėmelio linijos pavidalą bei spalvą, įrėminimo būdą.

Kontroliniai klausimai

1. Kuo skiriasi formulė ir funkcija *Microsoft Excel* programoje.
2. Ar galima funkciją panaudoti formulėje kaip formulės sudedamąją dalį?
3. Kada patogu naudoti sumavimo funkciją? Kiek argumentų gali turėti sumavimo funkcija?
4. Paaiškinkite vidurkio skaičiavimo funkciją. Kada patogu naudoti vidurkio funkciją?
5. Ar funkcijos argumentu gali būti kita funkcija?
6. Paaiškinkite loginę funkciją *IF*.
7. Ar į loginės funkcijos *IF* argumentus gali būti įterpta kita *IF* funkcija?

4.12.4. Atlyginimų apskaitos darbo lapo sudarymas

1. Darbo tikslai

Išmokti darbo lape sudaryti sąrašo tipo duomenų struktūras, atlikti skaičiavimus, pasitelkiant skaičiuoklės *MS Excel* formules ir funkcijas.

Susipažinti su darbo užmokesčio apskaitos algoritmu [26].

Dalykinis žaidimas

Reikės apskaičiuoti darbuotojams išmokamus atlyginimus. Darbuotojai priimti dirbti dviem būdais: pagal etatą ir valandininkai. Etatiniai darbuotojai dirba užimdami tam tikrą etatą, už kurį mokamas nustatytas pastovus mėnesinis atlyginimas. Valandininkams mokamas atlyginimas už idirbtas darbo valandas ir viršvalandžius. Už viršvalandžius darbuotojui mokama dvigubai.

Taigi darbuotojų atlyginimai gali būti apskaičiuojami dviem būdais:

- 1) Etatinių darbuotojų – pagal mėnesio tarifo atlygį;
- 2) Valandininkų – už dirbtą laiką ir viršvalandžius pagal valandos tarifo atlygį.

Šio darbo užduotis – priimti į grupę 6 darbuotojus, surinkti jų duomenis, apskaičiuoti atlyginimą užduotyje nurodytu laikotarpiu, išskaičiuojant būtinus mokesčius (pajamų, draudimo, ir kt.) pagal šiame praktiniame darbe pateiktą algoritmą.

2. Veiksmų eiga kuriant atlyginimų apskaitos darbo lapą

Ši darbą bus lengviau atlikti laikantis tam tikros veiksmų atlikimo eigos rekomendacijų, pasitelkiant skaičiuoklės *Excel* priemones.

1. Skaičiuoklės *Excel* pagrindiniame lange pagal 4.16 lentelėje pateikiamą pavyzdį sudarykite asmenų grupės darbo užmokesčio apskaitos darbo lapą *Sheet1*.

Darbo lapo 1–7 eilutėse įrašykite jo paskirties antraštę, t. y. dalykinio žaidimo įstaigos pavadinimą, darbo lapą sudarančio asmens pavardę, vardą ir grupę bei kitus rekvizitus. Remdamiesi 4.16 lentelėje pateikta struktūra suteikite laukams (stulpeliams) prasminius pavadinimus.

Kad būtų lengviau taikyti užduotyje naudojamas formules, laštelių koordinatės ir stulpeliai turėtų atitikti 4.16 lentelėje nurodytas koordinatas.

Rinkdami duomenis laikykitės tam tikrų reikalavimų. Įrašę duomenis į laštelę paspauskite <Enter>.

2. A1 laštelėje nustatykite ataskaitos datą, kuri atitiks kompiuterio laikrodžio duomenis. Tai atliekama iškviečiant funkciją žynį ir pasirenkant funkciją *NOW()*.

A1 laštelės duomenis formatuokite vykdydami komandą *Format* → *Cells*. Komandos *Number* kortelėje pažymėkite *Date* formatų grupę, o jos formatų sąrašė pasirinkite reikiamą datos formatą, pavyzdžiui, *yy-mmmm-dd* arba *dd-mmmm-yy*.

3. Grupės darbuotojų sąrašą pradėkite rinkti nuo B10 laštelės, įrašykite 6 asmenis. Darbuotojus surūšiuokite pagal pavardes abėcėlės tvarka.

4. Darbuotojams pasirinktinai nurodykite bankų kodus ir skirtingą kiekvieno darbuotojo sąskaitos numerį.

5. Nustatykite atlyginimų apskaitos mėnesį, duomenis įrašykite į stulpelį *Mėnuo*. Į A10 laštelę įrašoma pasirinktojo mėnesio pabaigos data, pavyzdžiui, 04 01 31, ir pasirenkamas atitinkamas datos formatas – *yy/mm/dd*. Vienodai užpildykite visų pasirinktų darbuotojų mėnesio darbo apskaitos datą, kopijuokite šiuos duomenis į A11:A15 laštelių bloką.

6. Grafoje *Tarifinis atlyginimas* įrašykite skirtingą darbuotojų atlygį (įvertinama valandininkų darbo valanda: litais už dirbtą valandą; etatinių darbuotojų – visas mėnesinis atlygis). Grafoje *Dirbta valandų* nurodykite valandininkų dirbtų valandų skaičių per mėnesį (pasirinkite 2–4 darbuotojus, kurie dirbs valandininkais). Dirbančiųjų pagrindiniu etatu dirbtų valandų skaičius nenurodomas.

7. Tarpinius duomenis išsaugokite kietajame diske ir diskelyje (diskelį būtinai patikrinkite antivirusinėmis programomis). Šių duomenų reikės tolimesniems praktiniams darbams atlikti.

8. Įkelkite failą ir prateškite lentelę įrašydami stulpelių pavadinimus, kaip pavaizduota 4.16 lentelėje.

9. Naudodamiesi skaičiuoklės *Excel* programos funkcijomis, duomenų formatais ir formulėmis, pagal šio darbo 3 skyriuje pateiktą algoritmą sudarykite grupės darbo užmokesčio apskaitos darbo lapą.

Pagal pateiktąjį algoritmą grafoje *Apskaičiuota suma* įrašykite kiekvieno darbuotojo nurodytojo mėnesio darbo užmokesčių (4.17 lentelė). Apskaičiuokite bendrą visų darbuotojų atlyginimą, sumą įrašykite į atitinkamą eilutės *Iš viso per mėnesį* ląstelę.

10. Išsaugokite duomenis *File* → *Save* komanda. Išanalizuokite gautus duomenis.

4.16 lentelė. Pradiniai darbo užmokesčio apskaitos lapo duomenys

A1 ląstelėje nurodant einamąją datą taikoma funkcija *NOW()*

=NOW()
Format-> Cells, Number, Date, dd-mmm-yy

1	A	B	C	D	E	F	G
1	17-Kov-04		ĮSTAIGOS PAVADINIMAS				
2		GRUPĖS UŽMOKESČIO APSKAITA					
3							
4		Ataskaitos autorius: V. Pavardė (grupė)					
5							
6							
7							
8	Mėnuo	Pavardė	Banko kodas	Sąskaita	Dirbta valandų	Viršvalandžiai	Tarifinis atlyginimas
9							
10	04 01 31	Aušraitė Dalia	6761	12149	1640	60	8,00 Lt
11	04 01 31	Augutis Kazys	6761	89321	1560	20	5,00 Lt
12	04 01 31	Babraitienė Jovilė	8938	10100			500,00 Lt
13	04 01 31	Jonaitis Petras	6763	98700	1640	40	4,00 Lt
14	04 01 31	Petraitis Giedrius	6763	35333	1640		21,56 Lt
15	04 01 31	Ramanauskas Jonas	6761	90154			180,00 Lt
16		Iš viso per mėnesį					
17		% nuo apskaičiuotos sumos					

A10 ląstelei suteikiamas datos formatas ir įrašoma data

04/01/31
Format-> Cells, Number, Date, dd mm yy

Kopijuojant duomenis į A11:A15 bloką, užpildomi visi darbuotojų mėnesio darbo apskaitos duomenys

Etatiniai darbuotojai

Valandininkai

4.17 lentelė. Mokesčių skaičiavimo duomenys

	H	I	J	K	L	M	N
1	MGL		Lengvata	Neapmok. Suma	MGL dalis	Apmok. suma	Mokėstis
2	200,00 Lt		1 lengvata	110,00 Lt	1,2	120,00 Lt	21,60 Lt
3			2 lengvata	100,00 Lt	1,8	180,00 Lt	33,60 Lt
4			3 lengvata	90,00 Lt	2,8	280,00 Lt	57,60 Lt
5			Be lengvatų	80,00 Lt	3,8	380,00 Lt	85,60 Lt
6							
7							
8	Apskaičiuota suma	Lengvata	Neapmok. suma	Pajamų mokėstis	Draudimas	Kiti atskaitymai	Išmokėtas atlyginimas
9							
10							
11							
12							
13							
14							
15							
16							
17							

$=1,1*SHS2$
 $=1*SHS2$
 $=0,9*SHS2$
 $=0,8*SHS2$

$=L2*SHS2$

$=M2*18\%$
 $=N2+(M3-M2)*20\%$
 $=N3+(M4-M3)*24\%$
 $=N4+(M5-M4)*28\%$

$=IF(E10>0;E10*G10+2*F10*G10;G10)$

$=SUM(H10:H15)$

3. Atlyginimo apskaitos algoritmo aprašymas

1. Nors E, F, G stulpelių duomenis galima parinkti laisvai, rekomenduojama laikytis tokių taisyklių:

- ✚ Trečdalis visų darbuotojų yra valandininkai, dalis iš jų dirba viršvalandžius, likusieji yra etatiniai darbuotojai;
- ✚ Valandininkų darbo valandos apskaičiuojamos taip:

$$h = 8 * n$$
,
 čia n – mėnesio dienų skaičius;
- ✚ Etatinių darbuotojų valandų skaičiaus nurodyti nereikia;
- ✚ Tarifinis valandininkų atlygis a gali būti skirtingas, atsižvelgiant į darbo pobūdį, ir įvairuoti nuo 4,00 Lt iki 150,00 Lt už 1 darbo valandą. Už viršvalandžius v mokama dvigubai. Šiame stulpelyje nurodomas etatinių darbuotojų mėnesinis atlyginimas. Šiuos tarifinius atlygius galima nurodyti priderinant prie konkretaus laikotarpio atlyginimų dydžio.

2. Grafos *Apskaičiuota suma* duomenys formulėje bus žymimi S ir apskaičiuojami pagal formulę:

$$S = \begin{cases} h*a + 2*v*a, & \text{jei } h > 0 \quad (\text{valandininkų}); \\ a, & \text{jei } h = 0 \quad (\text{etatinių darbuotojų}), \end{cases} \quad (3.1)$$

čia h – dirbta valandų (stulpelis E); v – viršvalandžiai (stulpelis G); a – tarifinis atlygis (stulpelis F).

Darbo lape surinkite darbo užmokesčio apskaitos formulę ir apskaičiuokite visų darbuotojų pirmojo mėnesio užmokestį. Šiuos duomenis surašykite į stulpelį *Apskaičiuota suma*. Norėdami surinkti šią formulę reikiamoje vietoje, pažymėkite ląstelę H10. Iškvietę funkciją žynį (fx), parinkite norimą funkciją, t. y. šiuo atveju reikalinga sąlyga įvertinanti funkcija IF (...).

Funkcijos IF(...) sintaksė:

$$=IF(\text{sąlyga}; \text{reikšmė}_1; \text{apskaičiavimo_formulė_jei_sąlyga_tenkinama}; \text{reikšmė}_2; \text{apskaičiavimo_formulė_jei_sąlyga_netenkinama})$$

$$=IF(E10>0; E10*G10+2*F10*G10; G10). \quad (3.2)$$

3. *Excel* programa reikia automatiškai apskaičiuoti kiekvienam darbuotojui skirtą sumą. Prisiminkite, kaip kopijuoti formulę į stulpelį ir eilutę. Pažymėkite bloką H10:H15 (H10 ląstelėje turi būti surinkta 3.2 formulė). Nurodykite *Edit* → *Fill*, *Down* komandą (taip formulė nukopijuojama ir perkeliama į stulpelį, apskaičiuojamus duomenis perduodant ir keičiant santykinėmis koordinatėmis pagal poslinkį).

4. Nustatykite bendrą visiems darbuotojams skirtą sumą.

Tuo tikslu pažymėkite H16 ląstelę ir surinkite formulę naudodami funkciją *SUM(skaičius1; skaičius2; ...)*, t. y. =SUM(H10:H15).

5. Išsaugokite duomenis. Jei yra galimybių, dalinę ataskaitą išspausdinkite.

Pajamų mokesčio apskaitos algoritmas

Pajamų mokesčio apskaitos algoritmas grindžiamas vienu pagrindiniu dydžiu – *MGL (minimaliu gyvenimo lygiu)*, kuris įvertinamas litais ir apskaičiuojamas pagal ekonominę šalies būklę ir kinta atsižvelgiant į lito infliaciją bei kitas priežastis.

Pajamų mokestis apskaičiuojamas remiantis Lietuvos Respublikos pajamų mokesčio apskaičiavimo įstatymu. Pateiksime ištraukas iš pajamų mokesčio apskaičiavimo įstatymo.

„5 *straipsnis*. Pagrindinėje darbovietėje gaunamų pajamų, susijusių su darbo santykiais, *neapmokestinamas minimumas* yra:

1. I grupės invalidams – 1,1 MGL (minimalaus gyvenimo lygio, nustatomo pagal Lietuvos Respublikos gyventojų pajamų garantijų įstatymą) pajamos (t. y. *1 lengvata*);
2. II grupės invalidams – 1 MGL pajamos (t. y. *2 lengvata*);
3. III grupės invalidams – 0,9 MGL pajamos (t. y. *3 lengvata*);
4. Kitiems šio straipsnio 1–3 punktuose nepaminėtiems asmenims – 0,8 MGL pajamos (t. y. *be lengvatų*)“.

„1 priedas. Pajamų mokesčio tarifai, pagal kuriuos apskaičiuojamas mokestis nuo pagrindinėje darbovietėje per mėnesį gautų su darbo santykiais susijusių apmokestinamųjų pajamų:

1. Apmokestinamųjų pajamų daliai iki 1,2 MGL imtinai taikomas 18 proc. tarifas.
2. Apmokestinamųjų pajamų daliai, viršijusiai 1,2 MGL, bet ne didesnei kaip 1,8 MGL, taikomas 20 proc. tarifas.
3. Apmokestinamųjų pajamų daliai, viršijusiai 1,8 MGL, bet ne didesnei kaip 2,8 MGL, taikomas 24 proc. tarifas.
4. Apmokestinamųjų pajamų daliai, viršijusiai 2,8 MGL, bet ne didesnei kaip 3,8 MGL, taikomas 28 proc. tarifas.
5. Apmokestinamųjų pajamų daliai, viršijusiai 3,8 MGL, taikomas 33 proc. tarifas“.

Pradedant mokesčių apskaitą mokesčių išskaičiavimo duomenys turi būti jau nustatyti (žr. 4.17 lentelę) taikant atitinkamas formules ir įrašyti konkrečiuose stulpeliuose J, K, L, M, N. Nustatoma tokia laukų struktūra: *Lengvata; Neapmokestinama suma; MGL dalis; Apmokestinama suma; Mokestis*.

Minimalus gyvenimo lygis – MGL įrašomas į H2 ląstelę ir yra lygus, pavyzdžiui, 200,00 Lt.

J2:N5 bloko informacija gaunama pagal formules užpildant atitinkamus stulpelių elementus.

M stulpelis užpildomas į M2 ląstelę įrašant formulę $=L2*SH\$2$ ir automatiškai užpildant stulpelį M2:M5; jis pažymimas ir nurodoma *Edit → Fill, Down* komandų grupė.

4.12.5. Pajamų mokesčio apskaita skaičiuojant atlyginimą

1. Darbo tikslai

Išmokti sudaryti darbo lapą, jame atlikti sudėtingesnius pajamų mokesčių skaičiavimus, naudojant skaičiuoklės *Excel* formules, funkcijas, duomenų formatus, grupių sumas.

Susipažinti su darbo užmokesčio apskaitos, pajamų mokesčio dydžio skaičiavimo algoritmu, nustatant pasirinktos darbuotojų grupės atlyginimus [26]. Sudaryti grupės darbo užmokesčio apskaitos darbo lapą, pagal kurį skaičiuojamos visos atskaitomos sumos ir išmokamas atlyginimas.

2. Darbo užduotys

Dalykinis žaidimas

1. Šio žaidimo metu naudosisimės anksčiau pradėto kurti užmokesčio apskaitos darbo lapo duomenimis: t. y. darbuotojų sąrašu, dirbtų valandų, tarifinio atlygio ir apskaičiuoto atlyginimo duomenimis. Jeigu šie duomenys saugomi diskelyje, būtinai jį patikrinkite antivirusinėmis programomis.

Reikės sudaryti formules pajamų mokesčiui apskaičiuoti atsižvelgiant į kai kuriems darbuotojams suteiktas lengvatas, nustatyti neapmokestinamų sumų dydį, apskaičiuoti darbuotojų *pajamų mokesčių*, atskaitomą *draudimo*, *kitų atskaitymų dydį* ir apskaičiuoti *išmokamą sumą*.

2. Atverkite darbo užmokesčio apskaitos lapą, sukurtą ir išsaugotą ankstesniųjų pratybų metu. Šią darbo lapo dalį turi sudaryti jau užpildyti duomenimis stulpeliai: pažymėtas *mėnuo*, už kurį atsiskaitoma, darbuotojų sąrašas (pvz., šeši darbuotojai), t. y. užpildyti laukai: *Pavardė*, *Banko kodas*, *Sąskaitos numeris*, *Dirbta valandų*, *Viršvalandžių skaičius* (valandininkų) ir jų darbo valandos *tarifų įkainiai*, arba etatinių darbuotojų viso tarifo atlyginimas. *Apskaičiuota suma* nustatoma pagal 3.1 formulę iš 4.12.4 praktinio darbo.

3. Pratęskite darbo lapą, užpildydami duomenimis laukus (t. y. stulpelių *I8:N8* pavadinimus ir duomenis), kaip parodyta 4.18 lentelėje.

Prieš rinkdami duomenis ir formules blokus, kurių duomenys bus išreiškiami piniginiiais vienetais, suformatuokite, t. y. suteikite *H10:H16, J10:N17* valiutos formatą – vykdykite *Format* → *Cells, Numbers, Currency* komandas ir pasirinkite tinkamą valiutos išraišką.

4. Pasirinkite darbuotojus, kuriems bus taikomos 1, 2 ir 3 mokesčių atskaitymo lengvatos. Lauke *Lengvata* nurodykite skaičius 1, 2, 3, t. y. lengvatų, kurios taikomos trims pasirinktiesiems jūsų sudarytos darbo grupės darbuotojams, tipus (dydis *l*).

5. Atitinkamai nustatykite neapmokestinamos sumos dydį, neapmokestinamą sumą *N*, priklausančią nuo lengvatos tipo *l*, apskaičiuodami pagal tokią formulę:

$$N = \begin{cases} N1, & \text{kai } l=1; \\ N2, & \text{kai } l=2 \\ N3, & \text{kai } l=3; \\ N4, & \text{kai lengvata netaikoma.} \end{cases} \quad (2.2)$$

Dydžius *N1, ..., N4* perrašykite iš praėjusiam darbe apskaičiuotos *neapmokestinamos sumos*, kuri buvo nustatoma pagal MGL dydį, atsižvelgiant į kiekvieną lengvatos tipą. Šių skaičių duomenys įrašyti į *K2:K5* (žr. 4.16.lentelę).

Norėdami apskaičiuoti *neapmokestinamos sumos* dydį, ląstelėje *J10* surenkame formulę, taikydami sąlygos įvertinimo funkciją *IF()*:

$$=IF(I10=1; \$K\$2; IF(I10=2; \$K\$3; IF(I10=3; \$K\$4; \$K\$5)))$$

Ją kopijuojame į *J10:J15* bloką, taikydami *Edit* → *Fill Down* komandą, kuri perskaičiuos formulę pagal santykinės ir absoliučias ląstelių koordinatas.

Atkreipkite dėmesį, kaip automatiškai užsipildė šis blokas. Jei pastebite nukrypimų, dar kartą patikrinkite, ar teisingai surinkote

formulę ir ar teisingai atlikote visas automatinio užpildymo komandas.

6. Apskaičiuokite darbuotojų *pajamų mokesį* (dydis P) pagal tokią formulę:

$$P = \begin{cases} M4+(S-N-R4)*33\%, & \text{kai } (S-N)>R4; \\ M3+(S-N-R3)*28\%, & \text{kai } R4 \geq (S-N) > R3; \\ M2+(S-N-R2)*24\%, & \text{kai } R3 \geq (S-N) > R2; \\ M1+(S-N-R1)*20\%, & \text{kai } R2 \geq (S-N) > R1; \\ (S-N)*18\%, & \text{kai } R1 \geq (S-N) > 0; \\ 0, & \text{kai } (S-N) \leq 0. \end{cases} \quad (2.3)$$

čia: P – pajamų mokestis, S – apskaičiuota suma, N – neapmokestinama suma;

$R1, R2, R3, R4$ – ribinės apmokestinamosios sumos, kai atskaitymų procentai skirtingi (šie dydžiai turi būti apskaičiuoti laštelėse M2:M5);

$M1, M2, M3, M4$ – ribinių apmokestinamųjų sumų pajamų mokesčiai (šie dydžiai apskaičiuoti laštelėse N2:N5 pagal praėjusio praktinio darbo duomenis, žr. 4.17 lentelę).

Tuo tikslu, darbo lape pažymėkite laštelę K10 ir surinkite joje pajamų mokesčio apskaitos formulę (žr. 4.18. lentelę):

$$\begin{aligned} &=IF(H10-J10>\$M\$5;\$N\$5+(H10-J10-\$M\$5)*33\%; \\ &IF(H10-J10>\$M\$4;\$N\$4+(H10-J10-\$M\$4)*28\%; \\ &IF(H10-J10>\$M\$3;\$N\$3+(H10-J10-\$M\$3)*24\%; \\ &IF(H10-J10>\$M\$2;\$N\$2+(H10-J10-\$M\$2)*20\%; \\ &IF(H10-J10>0;(H10-J10)*18\%;0)))) \end{aligned}$$

Kadangi daugelis formulės dalių kartojasi, tam tikras formulės dalis kopijuokite pasitelkdami *Edit* → *Copy*, *Paste* komandas. Formulę dar kartą atidžiai patikrinkite.

Šią formulę automatiškai kopijuokite į K10:K15 laštelių bloką, šį bloką pažymėkite ir vykdykite *Edit* → *Fill*, *Down* komandą – bus perskaičiuotos santykinės koordinatės.

7. Apskaičiuokite atskaitomą *draudimo* dydį D , kai draudimo procentas lygus 7.1. Lašelėje L10 surinkite formulę, skirtą pirmojo darbuotojo draudimo įmokai apskaičiuoti:

$$=H10*7,1\%$$

Vykdydami *Edit* → *Fill, Down* komandą, automatiškai kopijuokite formulę į L10:L15 bloką. Taip bus perskaičiuotos santykinės koordinatės ir gauti visų darbuotojų draudimo įmokų rezultatai.

8. Stulpelyje *Kiti atskaitymai* surinkite pasirinktinai dviejų arba trijų darbuotojų atskaitymų sumas, nustatytas, pavyzdžiui, esant koikiems nors išskolinimams. *Kiti atskaitymai* (žymimas dydis K) –tai iš atlyginimo išskaičiuojamos baudos, alimentai ir pan. Atskaitomas sumas pasirinktinai užpildykite stulpelyje M10:M15, nurodydami konkrečius dydžius valiuta (pvz., litais).

9. Apskaičiuokite *išmokamą atlyginimą*, t. y. dydį A , skaičiuojamą pagal formulę:

$$A = S - P - D - K,$$

čia S – *apskaičiuota suma* (šie duomenys pateikti stulpelyje H10:H15);

P – *pajamų mokestis* (šie duomenys pateikti stulpelyje K10:K15);

D – *draudimas* (šie duomenys pateikti stulpelyje L10:L15);

K – *kiti atskaitymai* (šie duomenys pateikti stulpelyje M10:M15).

Norėdami apskaičiuoti pirmojo darbuotojo atlyginimą, lašelėje N10 surinkite formulę:

$$=H10-SUM(K10:M10)$$

ir automatiškai užpildykite bloką N10:N15.

4.18 lentelė. Pajamų mokesčio apskaita

Pajamų mokestis:
 =IF(H10-J10>\$MS5;\$NS5+(H10-J10-\$MS5)*33%;
 IF(H10-J10>\$MS4;\$NS4+(H10-J10-\$MS4)*28%;
 IF(H10-J10>\$MS3;\$NS3+(H10-J10-\$MS3)*24%;
 IF(H10-J10>\$MS2;\$NS2+(H10-J10-\$MS2)*20%;
 IF(H10-J10>0;(H10-J10)*18%;0)))

Neapmokestinama suma:
 =IF(I10=1;\$KS2;IF(I10=2;\$KS3;
 IF(I10=3;\$KS4;\$KS5)))

Išmokėtas atlyginimas:
 =H10-SUM(K10:M10)

Draudimas:
 =H10*7,1%

	G	H	I	J	K	L	M	N
1		MGL		Lengvata	Neapmok. Suma	MGL dalis	Apmok. suma	Mokėstis
2		200,00 Lt		1 lengvata	110,00 Lt	1,2	120,00 Lt	21,60 Lt
3				2 lengvata	100,00 Lt	1,8	180,00 Lt	33,60 Lt
4				3 lengvata	90,00 Lt	2,8	280,00 Lt	57,60 Lt
5				Be lengvatų	80,00 Lt	3,8	380,00 Lt	85,60 Lt
6								
7								
8	Tarifinis atlyginimas	Apskaičiuota suma	Lengvata	Neapmok. suma	Pajamų mokestis	Draudimas	Kiti atskaitymai	Išmokėtas atlyginimas
9								
10								
11								
12								
13								
14								
15								
16	Iš viso per mėnesį							
17	% nuo apskaičiuotos sumos							

Apskaičiuota suma:
 =IF(E10>0;E10*G10+2*F10*G10;G10)

Iš viso per mėnesį:
 =SUM(H10:H15)

% nuo apskaičiuotos sumos:
 =J16*100/\$H16

10. Darbo lape apskaičiuokite darbuotojų grupės mėnesio rodiklius, t. y. B16 ląstelėje surinkite tekstą *Iš viso per mėnesį*, B17 ląstelėje surinkite tekstą *% nuo apskaičiuotos sumos*.

Apskaičiuokite stulpelių H, J, K, L, M ir N sumas. Tuo tikslu pažymėkite H16 ląstelę ir spustelėkite automatinės sumos mygtuką.

Naudodami mišriąsias koordinatas apskaičiuokite mokesčių procentus nuo apskaičiuotos sumos. Prisiminkite, kaip apskaičiuojami procentai. Pažymėkite J17 ląstelę ir surinkite formulę:

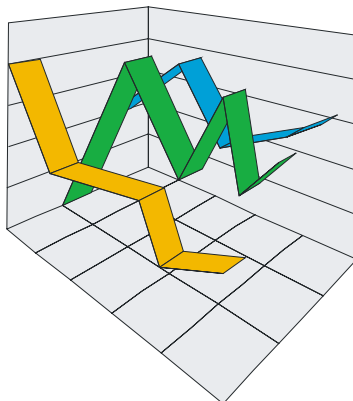
$$=J16*100/\$H16.$$

Kopijuokite formulę į J17:N17 bloką. Bloką reikia pažymėti ir automatiškai užpildyti vykdant *Edit → Fill, Right* komandą.

11. Naudodamiesi skaičiuoklės *MS Excel* programa sudarėte darbo užmokesčio apskaitos lapą. Duomenis išsaugokite. Jei yra galimybių, grupės užmokesčio apskaitos duomenis išspausdinkite.

Kontroliniai klausimai

1. Pademonstruokite ląstelėse surinktas formules, paaiškinkite surinktųjų formulių prasmę.
2. Kaip paaiškinsite atlyginimų apskaičiavimo algoritmo atlikimą *MS Excel* programos priemonėmis?
3. Pagal kokį principą skaičiuojamos lengvatos?



LITERATŪRA

1. Adomavičius J., Bareiša E., Jakutavičius A., Jusas V., Keršienė V., Lenkevičius A., Misevičienė R., Misevičius A., Motiejūnas K., Pečiukonienė M., Sekliuckis V., Sturienė R. *Informatika*. I dalis. Kaunas: Technologija, 1998.
2. Anthony R. N. *Planning and Control Systems: A framework for Analysis*. Harvard University Graduate School of Business Administration, 1965.
3. Booch G. *Object Oriented Analysis and Design with Applications*. 2nd edition. The Benjamin Cummings Publishing Co. Inc., 1994.
4. Breuker J. A., Widinga B. J. *KADS: Structured Knowledge Acquisition for Expert Systems*. Proceeding of Fifth International Workshop on Expert Systems and their Applications. Avignon, France, 1985.
5. Codd E. F. Extending the Database Relational Model to Capture More Meaning. *ACM Trans. Database Systems*. 1979. Vol. 4. No.4.
6. Čaplinskas A. *Programų sistemų inžinerijos pagrindai*. I ir II dalis. Vilnius: Matematikos ir informatikos institutas, 1996, 1998.
7. Dagys V., Denisovas V., Maciulevičius S., Otas A., Šakys V., Telešius E., Vidžiūnas A., Žandarys A. *Duomenų bazės*. ECDL projekto mokomoji medžiaga. Vilnius: Žara, 2000.
8. Dzemydienė D. Alternatyvių sprendimų analizė intelektualizuotose informacinėse sistemose. *Lietuvos mokslas ir pramonė*. Informacinės technologijos'99. Konferencijos medžiaga. Kaunas: Technologija, 1999.
9. Dzemydienė D., Naujikienė R. Informacinių sistemų mokymo problemos. *Lietuvos mokslas ir pramonė*. Informacinės technologijos'99. Konferencijos medžiaga. Kaunas: Technologija, 1999.
10. Dzemydienė D., Kažemikaitienė E., Petrauskas R. Unified Approach of Developing Advisory Information Systems in Crime Investigation. *Databases and Information Systems*. Proceedings

- of the Fift International Baltic Conference BalticDB&IS, ISBN 9985-894-5. Tallinn. 2002. Vol. 2.
11. Dzemydienė D., Naujikienė R. Informacijos sistemos. Patariamiosios sistemos teisėtvarkoje. *Informacijos mokslai*. Mokslo darbai Nr. 18. Vilniaus universitetas, 2001.
 12. Grigas G. *Duomenų tipai*. Vadovėlis. Vilnius: Žuvėdra, 1997.
 13. Haag S., Cummings M., Dawkins J. *Management Information Systems for the Information Age*. Irwin McGraw-Hill, 2000.
 14. *Information Technology and Society*. Edited by Nick Heap and others. London: SAGE Publications Ltd., 1996.
 15. Keras A., Petrauskas R. Informacinės technologijos teisės studijose. *Studijos aukštojoje mokykloje*. Konferencijos medžiaga. 1 knyga. Kaunas: Technologija, 1997.
 16. Leonenkov A. B. *Samoučitel UML*. Peterburg SPb.: BXV, 2002.
 17. Lucey T. *Management Information Systems*. 7th edition. London, 1995.
 18. Maskeliūnas S. Ontological Engineering: Common Approaches and Visualisation Capabilities. *Informatica*. 2000. Vol. 11. No. 1.
 19. McLeod R. Jr. *Management Information Systems*. Prentice Hall, 1998.
 20. Petrauskas R. Informacinių technologijų taikymas viešajame administravime. Vilnius, 2001.
 21. Saulis A., Vasilecas O. *Informacinių sistemų pagrindai*. Mokojoji knyga. Vilnius: Technika, 1998.
 22. Simanauskas L. *Kompiuterizuotos informacinės sistemos*. Vilnius, 1994.
 23. Solvberg A. Contents of Information Systems Engineering Education. In J. Barzdiniš (Ed.) *Databases and Information Systems*. Proceedings of Third International Baltic Workshop. Riga. 1998. Vol. 1.
 24. Starkus B., Kitkauskaitė A. *Microsoft Office 2000. Tai patogu ir paprasta*. Kaunas: Smaltija, 1999.
 25. Stern N., Stern R. *Computing in the Information Age*. John Wiley & Sons, 1993.

26. Šakys V. *Skaičiuoklė Excel 2000 ir 2002 firmos vadybai*. Vilnius, 2001, 2002.
27. Švanytė-Dzemydienė D., Naujikienė R. *Informatikos pagrindai. Šiuolaikiniai teksto tvarkymo būdai*. Vadovėlis. Vilnius: LTU, 2001.
28. *Teisinės informacijos kompiuterinė sistema LITLEX*. Aprašymas vartotojui. Teisinės informacijos centras. Vilnius. 2001.
29. Zeleznikow J. *Knowledge Discovery and Machine Learning in the Legal Domain*. Applied Computing Research Institute. Australia: La Trobe University, 1999. <http://www.austlii.edu.au>
30. Ziya Aktas A. *Structured Analysis and Design of Information Systems*. Prentice Hall International Editions, 1987.
31. Valavičius E., Stankevičius J., Leonavičius G. *Informatika*. Vadovėlis. Vilnius: Piktetas, 1999.

Doc. dr. Dzemydienė Dalė, Naujikienė Ramutė

Dz-08 Informacinės sistemos. Duomenų struktūros ir valdymas: vadovėlis. –
Vilnius: Lietuvos teisės universiteto Leidybos centras, 2004. – 264 p.,
28 lentelės, 202 paveikslai.

Bibliogr.: p. 261–263.

ISBN 9955-563-45-1

Vadovėlis skirtas aukštųjų mokyklų studentams, dėstytojams ir visiems besidomintiems informacinių sistemų kūrimo klausimais. Vadovėlyje pateikta teorinė medžiaga, metodiniai nurodymai bus naudingi būsimiesiems valdymo, administravimo, teisės specialistams ir visiems norintiems susipažinti su šiuolaikinėmis informacinių sistemų kūrimo galimybėmis. Praktinių darbų užduotys padės išsiugdyti gebėjimus dirbti su duomenų valdymo struktūromis ir informacinėmis sistemomis. Svarbu suprasti organizacijų aplinką ir susipažinti su pagrindinėmis duomenų išraiškos priemonėmis, modeliais bei valdymo ypatumais, užtikrinančiais efektyvesnius darbo metodus. Reikšmingą vaidmenį šiuolaikiniuose informacijos valdymo procesuose vaidina duomenų bazių valdymo sistemos, skaičiuoklių darbo aplinka, interneto technologijos. Jų teikiamos duomenų apdorojimo priemonės įgalina lengviau tvarkyti ir valdyti duomenis, juos išsaugoti, rūšiuoti, vykdyti paiešką, skleisti informaciją kompiuteriniuose tinkluose. Savarankiški įgūdžiai, įgyti atliekant praktines užduotis pagal šiame vadovėlyje pateiktus metodinius nurodymus, skatins taikyti modernias kompiuterines technologijas ir ieškoti naujų galimybių susipažinti su šiuolaikinėmis informacinėmis sistemomis.

UDK 004.42(075.8)

Dalė Dzemydienė, Ramutė Naujikienė
INFORMACINĖS SISTEMOS. DUOMENŲ STRUKTŪROS IR VALDYMAS
Vadovėlis

Redaktorė *Gitana Irmienė*
Rinkėja *Dovilė Gaidytė*
Maketuotoja *Regina Bernadišienė*
Viršelio dailininkė *Stanislava Narkevičiūtė*

SL 585. 2004 02 05. 11,69 leidyb. apsk. l.
Tiražas 1000 egz. Užsakymas .
Išleido Lietuvos teisės universiteto Leidybos centras, Ateities g. 20, 2057 Vilnius.
Tinklalapis internete www.ltu.lt
El. paštas leidyba@ltu.lt
Spausdino UAB „Baltijos kopija“, Kareivių 13b, Vilnius.
Tinklalapis internete www.kopija.lt
El. paštas info@kopija.lt