

Antanas Užusienis



MS ACCESS DUOMENŲ BAZIŲ AUTOMATIZAVIMAS

Automatizavimo pagrindai

2003 m.

Išleidžiama bendro kompiuterinio mokomojo leidinio (KLM) „*MS ACCESS* duomenų bazių automatizavimas. Automatizavimo pagrindai. Makrokomandos. Programavimas *VBA*“ pradžia - „*MS ACCESS* duomenų bazių automatizavimas. Automatizavimo pagrindai“

Tekstinę sritį sudaro pirma kompiuterinė knyga (103 psl.), objektų sritį - kartu komplektuojamų duomenų bazių pavyzdžių bei su pirma knyga siejamos papildomos medžiagos rinkinys.

Pradedančiųjų ir toliau pažengusiųjų kompiuterių vartotojų savimokai orientuotas kompiuterinis mokamasis leidinys sudarys sąlygas mokytis automatizuoti *Ms Access* duomenų bazes taikant įterptąsias programas (*Wizards*), makrokomandas ir programavimą *VBA* (*Visual Basic for Applications*).

Leidinyje pateikiami supažindinamieji teoriniai apibendrinimai ir aiškinamieji tekstai, apžvalginiai bendrosios paskirties ir specializuoti pavyzdžiai, pratimai, užduotys, iliustracijos, kontrolinės užduotys. Nagrinėjami įterptinių programų panaudojimo atvejai, suderintų (veikiančių) ir savarankiškam derinimui numatomų makrokomandų ir taikomųjų procedūrų arba programų tekstai ir jų aprašai, rečiau vartojamų ar naujesnių kompiuterinių terminų parankinis žodynas, ir atskirai duomenų bazių automatizavimo srities *VBA* programavimo aiškinamasis žodynas. Pateikiamos nuorodos kur ką galima rasti arba pakeisti ir kt. Aiškinamos komandos, jų panaudojimas. Daug kur greta angliško pavadinimo ir tekstų pateikiami lietuviški atitikmenys. Kompiuteriniai terminai tikslinti pagal kompiuterijos terminų žodynus.

Antanas Užsienis, 2003

TURINYS

Įvadas	5
1. Sistemos Access įterptinių programų taikymas duomenų bazėms automatizuoti.....	7
1.1. Duomenų bazės Expenses taikomosios programos tyrinėjimas.....	11
1.2. Kaip dirba specialiosios paskirties programa <i>Database Wizard</i>	15
1.3. Komandos mygtuko veikimo procedūros tyrimas	17
1.4. Paslaugų programos <i>Command Button Wizard</i> panaudojimas	20
1.5. Hipersaitai ir nuorodos	23
1.6. Paslaugų programa <i>Combo Box Wizard</i>	33
1.7. Paslaugų programa <i>Switchboard Manager</i>	35
1.8. Vartotojo sąsajų valdymas	37
1.9. Paleidimo parametrų nustatymas	38
1.10. Apsauga nuo nesankcionuotos kreipties	41
1.11. Komandų sąrašų ir mygtukų juostų parengimas	42
1.12. <i>Access</i> duomenų bazės automatizavimo rezultatų apžvalga	49
1.13. Programavimo taikymas duomenų bazėms automatizuoti.....	50
1.14. Duomenų valdymo kalbos.....	51
1.15. Santrauka	51
2. Objektai ir įvykiai.....	52
2.1. Objektinis mąstymo būdas	52
2.2. Objekto savybių aprašas.....	54
2.3. Konstravimo būsenos savybių apžvalga	54
2.4. Vykdomo būsenos savybių apžvalga	56
2.5. Paleisties parametrų nustatymas.....	57
2.6. Duomenų bazės savybių nustatymas.....	57
2.7. Sistemos <i>Access</i> aplinkos nustatymas	58
2.8. Apsaugos parametrų nustatymas	59
2.9. Įgaliojimų saugojimas objektuose <i>Document</i> ir <i>Container</i>	60
2.10. Objektų egzistavimo laiko trukmė	60
2.11. Objektų valdymo programos	60
2.12. Makroveiksmų komandų panaudojimas.....	61
2.13. <i>VBA</i> procedūrų panaudojimas	62
2.14. Sistemos <i>Access</i> įvykiai	62
2.15. <i>Access</i> programavimo modelis.....	63
2.16. Įvykių kategorijos.....	64
2.17. Objektams būdingos įvykių savybės	64
2.18. Duomenų atnaujinimo procesas	66
2.19. Nustatytosios standartinės elgsenos atšauktis	67
2.20. Santrauka	71
3. Sistemos <i>Access</i> objektinis modelis	73
3.1. Objektų sąsaja ir sąveika	73
3.1.1. Analogiškųjų objektų grupės.....	73
3.1.2. Sudėtiniai objektai.....	74
3.2. Sistemos <i>Access</i> architektūra	75
3.2.1. Sistemos <i>Access</i> programa <i>Application</i>	75
3.2.2. Duomenų bazės branduoliai <i>Jet</i> ir <i>MSDE</i>	76
3.2.3. Objektų hierarchija	77
3.3. Objektinis modelis <i>Access Application</i>	77
3.4. Objektinis modelis <i>Access Visual Basic</i>	78
3.5. Makroprogramavimo aplinka.....	78
3.5.1. Kreiptis į objektus, jų savybes, vardus.....	79
3.5.2. Nuorodos į pagalbinę formą.....	82

3.5.3. Nuorodos į veikiamųjų objektų valdomuosius elementus.....	84
3.5.4. Objekto <i>Screen</i> naudojimas nuorodai į veikiamąjį objektą.....	84
3.5.5. Nuorodos į lauką	85
3.6. Nuorodų rengimas komponuokliu <i>Expression Builder</i>	86
3.7. Santrauka	88
4. Veiksmai su formomis	88
4.1. Saitai tarp formų ir duomenų valdymo elementų.....	89
4.1.1. Įrašų rinkinio samprata	90
4.1.2. Formos arba ataskaitos įrašų šaltinis	93
4.1.3. Valdomųjų elementų duomenų šaltiniai	93
4.2. Šaltinio modelis „viena forma – vienas duomenų šaltinis“	95
4.2.1. Automatinis duomenų keitinys <i>AutoLookup</i>	95
4.3. Valdomųjų elementų ir laukų sąveika	96
4.3.1. Užklausos skaičiuojamieji laukai	97
4.4. Formos skaičiuojamieji laukai	97
4.4.1. Kaip gauti duomenis iš tos pačios formos.....	98
4.4.2. Kaip gauti duomenis iš kitos formos arba ataskaitos	100
4.4.3. Duomenų gavimas iš kito šaltinio funkcija <i>DLookup()</i>	101
4.5. Nesietinis elementas kaip kintamasis	102
4.5.1. Priverstinis valdomojo elemento duomenų gavimas.....	103
4.5.2. Formos globaliųjų kintamųjų naudojimas.....	103
4.6. Valdomieji elementai su dviem duomenų šaltiniais.....	103
4.6.1. Duomenų paieškos kombinuotasis blokas.....	103
4.6.2. Duomenys nesietinam elementui panaudojant bloko savybę <i>Column</i>	105
4.6.3. Duomenys siejamam valdomajam elementui.....	106
4.7. Formų tarpusavio sąveika.....	106
4.7.1. Dviejų formų sinchronizavimas	106
4.7.2. Sinchronizavimas pagrindinės ir pagalbinės formos.....	109
4.8. Santrauka	110

Ivadas

Duomenų bazių (duomenų bankų) dažniausi pagrindiniai objektai yra: lentelės (*Tables*), užklausos (*Queries*), duomenų formos (*Forms*), ataskaitos (*Records*), su sistema *Access 2000* ir *XP* prisidėjo dar ir duomenų puslapiai (*Pages*). Tačiau, be automatizuotam valdymui skirtų pagrindinių objektų: makrokomandų (*Macros*) ir programų modulių (*Modules*), duomenų bazė yra sukuriamą ir valdomą tik dialoginiu būdu. Norint parengti ir valdyti tokios neautomatizuotos duomenų bazės duomenis vartotojui reikia žinoti daug komandų ir mokėti dirbti rankiniu būdu. Tuo atveju produktyvus ir efektyvus gali būti tik kvalifikuotas vartotojas.

Panaudojant duomenų bazės automatizavimo pagrindinius objektus – makrokomandas (*Macros*), ir modulių (*Modules*) programas, galima labai palengvinti vartotojo darbą, nes eiliniam uždaviniui įvykdyti kartais pakanka spustelėti, tik vieną kitą klavišą ar mygtuką. Tačiau automatizavimo objektams parengti reikia mokėti programuoti, turėti papildomų žinių ir daug dirbti.

Microsoft Access sistema turi numatytas priemones makrokomandoms (makrokomanda – elementarių komandų seka) rengti netaikant programavimo. Be to, sistema *Access* savo sudėtyje turi specializuotas programas - vedlius (*Wizards*), taikomosioms programoms panaudoti ir papildyti. Specializuotos įterptosios paslaugų programos padeda automatizuoti duomenų bazių rengimą, redagavimą ir duomenų valdymą. Be abejo, makrokomandas galima ir programuoti. Tačiau jų galimybės yra ribotos. Paprogramės, kurios vadinamos procedūromis (*Procedure*), ir duomenų bazių taikomosios programos (dažniausiai – procedūrų ir funkcijų sekos), kitaip vadinamos prietaikomis (*Application*), gali būti programuojamos taikant kurią nors objektinę (*Object Oriented*) programavimo aplinką, pvz.: *Borland C++*, *Delphi* ir kt.

Šiame mokomajame leidinyje aiškinama ir mokoma programuoti panaudojant korporacijos *Microsoft* sistemoje *Access* integruotą programavimo aplinką *Visual Basic for Applications (VBA)*, skirtą duomenų bazių objektams kurti ir duomenims valdyti. Jos pagrindas yra objektinė programavimo kalba *Visual Basic (VB)*. Programavimui *VBA* reikalingos programos nei pirkti, nei atskirai įdiegti atskirai nereikia - ji įdiegiama kartu su sistema *Access* ir automatiškai parengiama programavimo darbams kiekvienos duomenų bazės įkrovimo metu. Reikia tik išmokti ją atrasti ir programuoti.

Tenka priminti, kad *VBA - Visual Basic for Applications* - programų (sistemų) rinkinio *Microsoft Office 97* ir ypač *Microsoft Office 2000*, *Microsoft OfficeXP* pagrindinėse taikomojo pobūdžio sistemose *Word*, *Excel*, *PowerPoint* ir *Access* integruota suvienodinta *VBA* aplinka su panašiomis priemonėmis, panašiais darbo principais bei panašia programavimo kalbos sintakse. Galima teigti, kad *VBA* integruotoji aplinka, pagrįsta *Visual Basic 6.0*, dabar yra panaši visose *Ms Office 2000* ir *Ms Office XP* pagrindinėse sistemose. *VBA* kalbos siktakę sąlygoja tik pačios taikomosios sistemos, pvz., *Access* apdorojamų duomenų pobūdis bei specifika.

(Norintiems mokytis *VBA* programavimo pagrindų savimokos būdu rekomenduojama knyga, kurią 2003 m. išleido „Smaltija“:

A.Užusienis, „Programavimas VBA aplinkoje: Ms Word ir Excel“.

Apibendrinant galima sakyti, kad vartotojas duomenų bazės rengimą, duomenų objektų ir duomenų valdymą bei panaudojimą gali vykdyti pasirinkdamas arba rankinį – dialoginį/interaktyvųjį veikimą, t.y. be programavimo, arba tinkamai panaudoti sistemos *Access* įterptąsias taikomasias programas (*Wizards*) ir kartu savo iniciatyva taikyti makrokomandų rengimą (makroprogramavimą) ir programavimą *VBA*.

Programuotojai tvirtina, jog taikant programavimą galima beveik visiškai automatizuoti duomenų bazę (išskyrus kai kurią pradinį rengimą ir redagavimą, pvz., pradinių duomenų įvedimą) ir labai palengvinti vartotojo darbą.

PASTABOS.

Čia ir žemiau: pagrindiniai objektai - tai duomenų bazės struktūros pagrindinės sudedamosios dalys, kurios duomenų bazės pagrindiniame lange vadinamos objektų kortelių vardais (pvz.: *Tables, Queries, Forms* ir kt.); duomenų bankas = duomenų bazė (DB); rinkmena = failas; aplankas = katalogas; sistema = pagrindinė programa, pvz., *Microsoft Access*; įterptinė programa = vidinė taikomoji pagalbinė programa arba tiesiog prietaika; kodas = programa; žiniatinklis = tinklalapis. Vartotojas ras ir naujų žodžių, taikomų ir aiškinamų pagal kontekstą: visybė, užvaldomumas, duomenų puslapiai ir kt.

KAIP MOKYTIS, KAI VARTOTOJAS JAU TURI, KOMPAKTINĮ DISKĄ IR TINKAMĄ KOMPIUTERĮ?

Reikėtų atsižvelgti į du dalykus:

- vartotojo pradinį pasirengimą bei poreikį ir iš anksto numatomą gylį studijuoti duomenų bazes (ką jau moki ir ką dar nori arba reikia mokėti);
- mokomojo leidinio ir kompaktinio disko struktūrą (kada, kur ir ką sau tinkamo greitai rasti).

Jeigu vartotojas tik pradeda dirbti su duomenų bazėmis, lygiagrečiai jau galima pradėti domėtis galimais automatizavimo veiksmais ir nagrinėti pirmąją mokomojo leidinio dalį. Jeigu jau dirba, moka dialoginiu būdu sukurti elementarius duomenų bazės objektus ir tvarkyti bei surasti reikiamus duomenis, siūloma nagrinėti automatizavimą be programavimo, t. y. pirmąją ir antrąją mokomojo leidinio knygą. Jeigu skaitytojas moka kurti ir valdyti duomenų bazės svarbesnius objektus, be to, nori ir jau turi pradinį pasirengimą (gali arba reikia) mokytis ir mokėti automatizuoti duomenų bazes taikant programavimą, galima iš karto pradėti nagrinėti trečiąją knygą su išsamiai aiškinamais pratimais ir, esant poreikiui, „ekskursuoti“ į antrąją arba ir į pirmąją dalį, kuri visuomet „po ranka“ – kompaktiniame diske arba kompiuterio applanke.

Visas mokomojo leidinio knygas vartotojas ras įrašytas viename kompaktiniame diske (arba po vieną atskiruose kompaktiniuose diskuose):

Be to, kompaktiniame diske pateikiama:

- trumpas duomenų bazės pavyzdžio *Northwind.mdb* aprašas ir duomenų bazių objektų programavimo rezultatų pavyzdžiai pagal knygų ir skyrių pratimus, užduotis ir parengtas programas;
- papildoma informacija, reikalinga programuotojui (objektai, metodai, savybės konstantos, įterptosios funkcijos, operatorių pavyzdžiai su aiškinamaisiais teksta ir t.t.);
- duomenų bazių pavyzdžiai, rasti internete, ir kiti duomenys;
- įvairaus sudėtingumo programų pavyzdžiai, tinkami tyrinėti ir taikyti vartotojo programose ir projektuose iš konsultacijų (*Help*) temų;
- kita informacija.

1. Sistemos Access įterptinių programų taikymas duomenų bazėms automatizuoti

- Duomenų bazių rengimas programa *Database Wizard*
- Operacijų automatizavimas programa *Command Button Wizard*
- Kombinuotųjų sąrašų rengimas programa *Combo Box Wizard*
- Perkėlimo į reikiamos informacijos vietas organizavimas hipernuorodomis
- Komutacinio skydelio modifikavimas programa *Switchboard Manager*
- Vartotojo sąsajų valdymas
- Komandų sąrašų ir mygtukų darinių automatizuotas rengimas
- Programavimo prielaidos, paskirtis ir priemonės

Pagrindinė sistema *Microsoft Access* savo sudėtyje turi ir gali pateikti įterptines (vidines) taikomąsias programas/prietaikas duomenų bazėms arba duomenų bazių objektams sukurti ir duomenims valdyti vartotojui savo iniciatyva netaikant jokio programavimo ir net neturinčiam menkiausio supratimo apie programavimą. Tuo atveju vartotojas pasirenka, sužadina ir valdo vidines (nematomas) programas, lyg iš išorės, nurodydamas komandas įprastu būdu.

Duomenų bazių automatizavimo galimybių nagrinėjimą pradėsime įterptinės specialios paskirties programos/paslaugų programos *Database Wizard* analize.

Duomenų bazių vedlys *Database Wizard* gali automatizuotai sukurti šablonus tokių domenų bazių:

- *Asset Tracking*, turto apskaita,
- *Contact Management*, kontaktų valdymas,
- *Event Management*, įvykių valdymas,
- *Expenses*, sąnaudų apskaita,
- *Inventory Control*, sandėlio valdymas,
- *Ledger*, finansų apskaita,
- *Order Entry*, paraiškų apskaita,
- *Resource Scheduling*, išteklių planavimas,
- *Service Call Management*, aptarnavimo valdymas,
- *Time and Billing*, sąskaitų apmokėjimo planavimas.

Pagal pavadinimą pasirinkus dominančią duomenų bazę, sistemos *Access* specializuota programa – duomenų bazės vedlys (*Database Wizard*) sužadins taikomąją programą duomenų bazei automatizuotai projektuoti bei valdyti ir sukurs duomenų bazės šabloną/trafaretą, kurį po to bus galima modifikuoti ir pritaikyti sau dialoginiu būdu. Duomenų bazės rengimo pradžioje vartotojui reikės tik šiek tiek įsiterpti – pasirinkti ir nustatyti kai kuriuos pradinis parametrus, o toliau - tik laukti ir stebėti, kaip vykdomas progresyvus automatizuotas duomenų bazės projektavimas.

PRATIMAS. DUOMENŲ BAZĖS *EXPENSES* RENGIMAS IR TYRINĖJIMAS.

1. Paleiskite veikti *Microsoft Access*.
2. Pagalbiniam dialogo lange *Microsoft Access* įjunkite jungiklį *Access database wizards, pages, and projects* ir bakstelėkite *OK*.
3. Dialogo lange *New* bakstelėkite kortelės *Databases* auselę, pažymėkite rinkmeną *Expenses*. (Rinkmenos visą vardą su plėtiniu *Expenses.mdz* matysite, kai spustelėsite pelės dešinįjį klavišą ir iš parankinio komandų sąrašo nurodysite *Properties*.) Automatizuotu būdu sukurti duomenų bazei *Expenses* (duomenų bazės šablonui) scenarijaus instrukcija (taikomoji programa) yra įrašyta rinkmenoje *Expenses.mdz*. Bakstelėkite *OK*.
4. Dialogo lange *File New Database*, aplankų/katalogų grupėje *My Documents*, sukurkite aplanką *Macro&VBA* ir jame išsaugokite rengiamą duomenų bazę vardu *DB_EXPENSES* (įvesties laukelyje *File Name* surinkite *DB_EXPENSES*. Duomenų bazės rinkmenos vardo plėtinį *.MDB* sistema *Access* pridės pati). Bakstelėkite mygtuką *Create*.
5. Pirmame dialogo lange *Database Wizard* pranešama, kokie duomenys duomenų bazėje *Expenses* bus saugomi *The Expenses database will store*:
 - Information about employees, tarnautojų duomenys
 - Expense report information, sąnaudų ataskaitos
 - Expense Details, sąnaudų punktai
 - Expense Categories, sąnaudų kategorijos

Žemiau priminimas: *Click Next to continue*.

Bakstelėkite mygtuką *Next*.

6. Kitame *Database Wizard* dialogo lange yra pateikiami duomenų bazės lentelių pavadinimai *Tables in the database* ir lentelių laukai *Fields in the table*.

Duomenų bazės *DB_EXPENSES* lentelių pavadinimai *Tables in the database* ir lentelių laukai *Fields in the table*, matomi dialogo *Database Wizard* lange:

1 lentelė

Eil. Nr.	Lentelių pavadinimai <i>Tables in the database</i>	Lentelių laukai <i>Fields in the table</i>
1.	<i>Information about employees, tarnautojai</i>	<i>Department Name</i> darbovietė <i>Employee ID</i> , identifikatorius <i>Social Security Number</i> , draudimo numeris <i>Employee Number</i> , tarnautojo numeris <i>First Name</i> , vardas <i>Last Name</i> , pavardė <i>Email Name</i> , el. pašto adresas <i>Extension</i> , papildymas <i>Address</i> , adresas <i>City</i> , miestas <i>State/Province</i> , valstija, provincija <i>Postal Code</i> , pašto kodas <i>Country</i> , šalis <i>Home Phone</i> , namų telefonas <i>Work Phone</i> , darbo telefonas

		<i>Birthdate</i> , gimimo data <i>Date Hired</i> , įdarbinimo data <i>Notes</i> , pastabos
2.	Expense report information, sąnaudų ataskaitos	<i>Expense Report ID</i> , ataskaitos identifikatorius <i>Employee ID</i> , tarnautojo identifikatorius <i>Expense Type</i> , sąnaudų tipas <i>Exp Rpt Name</i> , sąnaud ataskaitos pavadinimas <i>Exp Rpt Descr</i> , sąnaudų aprašas <i>Date Submitted</i> , pateikties data <i>Advance</i> , paskola <i>Department Charged</i> , įstaigos skola <i>Business Purpose</i> , verslo tikslas <i>Paid</i> , išmokos
3.	<i>Expense Details</i> , sąnaudų punktai	<i>Expense Detail ID</i> , sąnaudų punkto identifikatorius <i>Expense Report ID</i> , sąnaudų ataskaitos identifikatorius <i>Expense Category ID</i> , sąnaudų kategorijos identifikatorius <i>Expense Item Amount</i> , sąnaudų punkto suma <i>Expense Item Description</i> , sąnaudų punkto aprašas <i>Expense Date</i> , sąnaudų data
4.	<i>Expense Categories</i> , sąnaudų kategorijos	<i>Expense Category ID</i> , sąnaudų kategorijos identifikatorius <i>Expense Category</i> , sąnaudų kategorija <i>Expense Account#</i> , sąnaudų suma

Papildomus laukus, kurių vardai dialogo lango *Database Wizard* lauke *Fields in the table* parašyti kursyvu, pagal reikalą, galima pažymėti, tada jie bus įtraukti į lentelę. Taip ir padarykite. Peržiūrėkite visas 4 lenteles, jų laukus. Dirbdami su *Access 97* lango apačioje įjunkite jungiklį *Yes, include sample data*, vaizduoti duomenų bazę su pavyzdžiais, (gaila, kad *Access 2000* to nėra). Bakstelėkite mygtuką *Next*.

7. Naujame dialogo lange peržiūrėkite ir pasirinkite ekrano foną, pvz., *Standard* ir bakstelėkite *Next*.

8. Kitame dialogo lange pasirinkite ataskaitų pateikties stilių, pvz., *Corporate* ir bakstelėkite *Next*.

9. Priešpaskutiniame dialogo lange yra galimybė įsikelti paveikslėlį arba galima nieko nekeisti. Jeigu jau yra išvestas duomenų bazės pavadinimas *Expenses*, bakstelėkite *Next*.

10. Paskutiniame dialogo lange *Database Wizard*, jau įjungtas jungiklis: *Yes, start the database*, meistras sukurs duomenų bazės šabloną, kuris bus išvestas ekrane. Galima įjungti jungiklį: *Display help on using a database*, rodyti aiškinimus, kaip panaudoti duomenų bases. Bakstelėkite *Finish*.

Pastaba: Jeigu prie kompiuterio neprijungtas spausdintuvas, *Database Wizard* atsisakys sukurti duomenų bazės šabloną.

Jeigu nepasirodys pranešimų apie nesklandumus, kai bakstelėsite *Finish*, ekrane matysite duomenų bazės (šablono) rengimo procesą. Pirmiausia (tačiau tik akimirkai) parodomas modulių langas su komandomis. Po to *Database Wizard* parengia lenteles, formas ir ataskaitas. Netrukus

duomenų bazės šablono rengimas baigiamas (rengimo trukmė priklauso nuo kompiuterio pajėgumų).

Jeigu paskutiniame dialogo lange *Database Wizard* įjungėte jungiklį: *Display Help on using a database*, galėsite paskaityti aiškinimą. Kitu atveju ekrane matysite duomenų bazės valdymo skydelio/komutatoriaus langą *Main Switchboard* (1.01 pav.).

Komutatorius – *switchboard* – įtaisas dviem ar daugiau įtaisų (objektų) tarpusavyje susieti. Trumpai apžvelkime tai, ką sukūrė sistema *Access* beveik be vartotojo pastangų. Specialios paskirties programa *Database Wizard* pagal duotąjį veiksmų scenarijų (*expenses.mdz*) programavimo priemonėmis *VBA* ir kt. automatiškai programavo ir koregavo taikomąją programą duomenų bazės šablonui *DB_EXPENSES* parengti bei valdyti - parengė rinkmeną *expenses.mdb*.

Parengtą programą sistema *Access* tuoj pat įvykdė, ir jos veikimo rezultatas akivaizdus - dabar jau matomas kompiuterio ekrane tik ką suprojektuotas duomenų bazės šablonas.

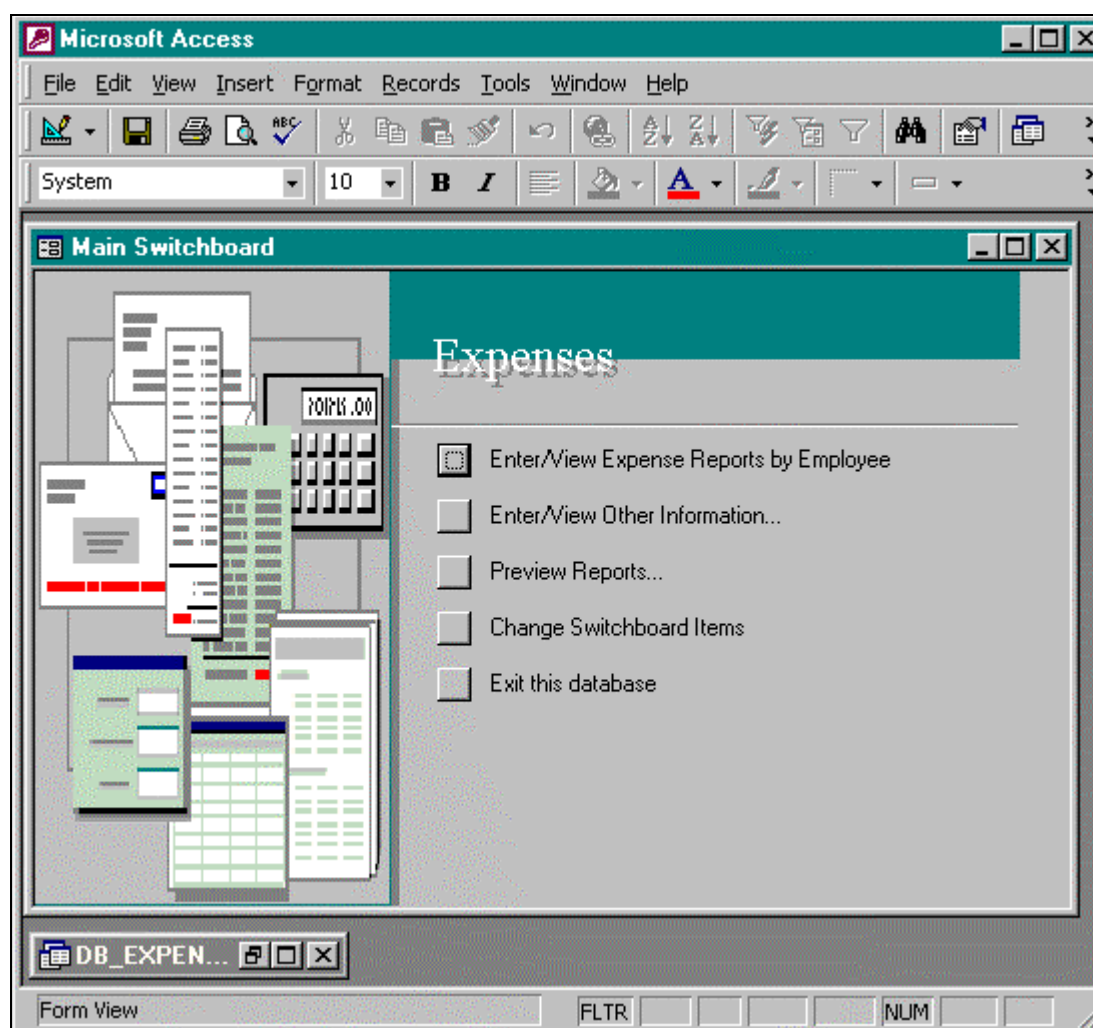
Vartotojas duomenų bazės šablono turinį, veikimą ir valdymą gali keisti/redaguoti savo nuožiūra, žinoma, atsižvelgdamas į sistemos galimybes bei ribotumus ir savo išprusimą.

Dabar jau galima nagrinėti specializuota programa *Database Wizard* automatizuotai parengta duomenų bazės *Expenses* taikomąją programą.

1.1. Duomenų bazės Expenses taikomosios programos tyrinėjimas

Duomenų formos lange *Main Switchboard* yra pateiktas pagrindinis valdymo/komutacinis skydelis *Expenses* su komandiniais mygtukais(1.01 pav.):

- Enter/View Expense Reports by Employee, sąnaudų ataskaitos pagal tarnautojus.
- Enter/View Other Information, kita informacija.
- Preview Reports, ataskaitų peržiūra.
- Change Switchboard Items, komutacinio skydelio punktų keitimas.
- Exit this database, darbo pabaiga.



1.01 pav. Duomenų bazės *DB_EXPENSES* valdymo centro – komutatoriaus *Main Switchboard* rodinys sistemos *Access* pagrindiniame lange

Formos *Main Switchboard* pirmasis komandinis mygtukas *Enter/View Expense Reports by Employee* atveria sudėtinės formos langą *Expense Reports by Employee*. Pagrindinėje formos dalyje (viršuje) galima matyti tris įrašus, jeigu jie buvo automatiškai suformuoti (pvz., *Access 97*) arba įrašus suformuosite patys (žr. žemiau 1.01 UŽDUOTIS; vykdydami užduotį įterpsite keletą įrašų, neskirdami ypatingo dėmesio jų turiniui, juos dar tikslinsite vėliau, dabar vėl peržiūrėkite duomenų bazės objektus). Pagalbinėje formoje (apačioje) pateikiami pagal tarnautojus sąnaudų įrašai.

Panaudokite lango apatinėje eilutėje *Record* esančius kryptinių mygtukus ir peržiūrėkite įrašus.

Kai bakstelėsite vardinį mygtuką *Expense Report Form*, ekrane matysite vieno įrašo formos *Expense Reports* langą taip pat susidedantį iš pagrindinės ir pagalbinės dalies. Ataskaitą spausdinimo lape peržiūrėsite, kai formoje bakstelėsite vardinį mygtuką *Preview Report*. Jeigu forma tuščia (be įrašų) matysite standartinį pranešimą: *Enter employee before entering expence report*, įvesk tarnautoją prieš atveriant ataskaitą.

Iš pabaigos į valdymo skydelio langą grįšite, kai uždarysite tris išvestuosius langus.

Atlikote vienkryptį pasižvalgymą. Grįžti atgal galėsite bet kada, tačiau tik uždarydami prieš tai išvestą langą, nes grįžties mygtukų peržiūrėtuose languose nėra.

The screenshot shows the Microsoft Access application window with the 'Main Switchboard' and 'Expense Reports by Employee' form. The form is titled 'Expenses' and contains the following fields:

- First Name: Davolio
- Last Name: Salesperson
- Title: 65432
- Employee #: Nancy
- Expense Report ID: 1
- Date Submitted: 1995.03.01
- Employee Name: Davolio Salesperson
- Advance: 0,00 Lt
- Exp Rpt Name: Feb. '95 Sales Trip
- Expense Totals: 464,00 Lt
- Exp Rpt Descr: Expenses during sales trip.
- Balance Due: 464,00 Lt
- Business Purpose: (empty)
- Dept. Charged: (empty)
- Paid: ☐

At the bottom of the form, there is a table with the following data:

Expense ID	Expense Category	Description	Amount
1995.02.01	Transportation	Plane ticket	431,00 Lt
1995.02.02	Meals	Breakfast meeting with Tomm	33,00 Lt

Below the table is a 'Preview Report...' button. The status bar at the bottom indicates 'Record: 1 of 1'.

1.02 pav. Vienos krypties maršruto objektų rodiniai: iš pagrindinio valdymo skydelio *Main Switchboard* atveriamą formą *Expense Reports by Employee*, sąnaudos pagal tarnautojus, po to formoje bakstelint mygtuką *Expense Report Form* - atveriamą formą *Expense Reports*, sąnaudų pateiktys, kurioje bakstelint mygtuką *Preview Report* – atveriamą ataskaitą *Expense Report*, sąnaudų apskaita (žr. žemiau ataskaitos rodinį 1.03 pav.)

Expense Report

Expense Report

Date Submitted

Employee ID

Employee Name

Exp Rpt Name

Dept. Charged

Description

Business Purpose

<i>Date</i>	<i>Description</i>	<i>Expense Category</i>	<i>Amount</i>
1995.02.01	Plane ticket	Transportation	431,00 Lt
1995.02.02	Breakfast meeting with Tomm	Meals	33,00 Lt

Total Expenses 464,00 Lt

Advance Amount 0,00 Lt

Balance Due 464,00 Lt

Page:

1.03 pav. Objektų apžvalgos vienakrypčio maršruto tęsinys: formoje *Expense Reports* bakstelint mygtuką *Preview Report* atveriamą ataskaitą *Expense Report*, sąnaudų apskaita

Atkreipkite dėmesį į tai, kad sinchronizuojami duomenys pagrindinėje ir pagalbinėje formoje, taip pat formoje ir ataskaitoje.

Sinchronizacija – synchronization, timing – dviejų ar kelių procesų ar reiškinių laikinis suderinimas, kad kiekvienas jų vyktų nustatyto sąryšio tvarka.

Main Switchboard antrasis komandinis mygtukas *Enter/View Other Information* atveria kitą - pagalbinio valdymo/komutacinio skydelio langą *Forms Switchboard*, kuriame valdymo mygtuku *Enter/View Expense Categories* galima atverti duomenų įvesties formą *Expense Categories* arba, kai bakstelėsite kitą mygtuką *Return Main Switchboard*, grįšite į pagrindinį langą *Main Switchboard*.

Duomenų bazės lange *Microsoft Access - [DB_EXPENSES:Database]* pagrindinį komutacinio skydelio langą (*Main Switchboard*) galėsite išvesti, kai atversite objektų *Forms* kortelę, pažymėsite *Switchboard* ir bakstelėsite mygtuką *Open*.

Main Switchboard trečiasis komandinis mygtukas *Preview Reports* atveria dar vieną pagalbinį komutacinį skydelį *Reports Switchboard*, kuriame yra trys komandiniai mygtukai:

- Preview the Expense Rpt Summary by Category Report
- Preview the Expense Rpt Summary by Employee Report
- Return to Main Switchboard.

Pirmasis ir antrasis komandinis mygtukas atveria dialogo tipo specialią formą ataskaitos pradžios ir pabaigos datoms įvesti. Po to, vardiniu mygtuku *Preview* galima peržiūrėti nurodyto laiko intervalo sumines ataskaitas. Tai panašu į parametrinių užklausų veikimą nors duomenų

bazės lange jokių užklausų *Queries* nėra. Todėl galima spėti, kad užklausas suformuoja procedūros (paprogramės).

Pagrindinio *Main Switchboard* ketvirtasis komandinis mygtukas *Change Switchboard Items* atveria komutacinio skydelio valdiklio langą *Switchboard Manager*, kuriame pateikiamas redagavimo mygtukų stulpelis ir išvardyti trys jau žinomi komutaciniai skydeliai – *Switchboard Pages*:

- Main Switchboard (Default)
- Forms Switchboard
- Reports Switchboard.

Apie galimus redagavimo veiksmus kalbėsime vėliau, dabar uždarykite *Switchboard Manager* langą – *Close*.

Komutatoriaus (*Main Switchboard*) penktasis komandinis mygtukas uždaro duomenų bazės langą. Duomenų bazės komutacinio skydelio langą vėl matysite ekrane, jeigu sistemos *Microsoft Access* lange atversite komandų sąrašą *File* ir komandų sąrašo apatinėje dalyje pele bakstelėsite duomenų bazės rinkmenos vardą.

Atlikite užduotis. Pradedantiesiems rekomenduojama viena suprastinta 1.1 užduotis be duomenų bazių pertvarkymo/konvertavimo.

1.01 UŽDUOTIS.

1. Kopijuoti į savo darbo aplanką, pvz., *VadovasIkn_VBA* iš kompaktinio disko „*ACCESS DB automatizavimas*“ aplanko *MSO-97* sukurtą sistemą *Access 97* ir užpildytą keliais įrašais duomenų bazę *Expenses.mdb*.
2. Pertvarkyti/konvertuoti sistemą *Access 97* sukurtą duomenų bazę *Expenses.mdb* į sistemą, su kuria dabar dirbama, pvz.: *Access 2000* arba *Access XP* ar kurią nors kitą. (Jeigu pradedančiam vartotojui duomenų bazės (DB), sukurtos ankstesne *Access* atmaina pertvarkymas/konvertavimas į dabartinę sistemą per daug sudėtingas ar nesuprantamas, iš kompaktinio disko galima kopijuoti duomenų bazę *Konvert-Expenses.mdb*, kuri jau pertvarkyta į *Access 2000*).
3. Tyrinėti DB nuosekliai nuo 1.1 skyrelio pradžios.

Priminimas: prieš DB konvertavimą išjunkite rinkmenos požymio/atributo *Read-only*, tik skaityti, jungiklį: *File* → *Properties*; pagrindiniame sistemos *Access* lange nurodykite: *Tools* → *Database Utilities* → *Convert Database* → *To Current Access Database Version*; dialogo lange *Database to Convert From* pasirinkite kokią DB konvertuoti; kitame lange *Convert Database Into* įrašysite naują DB vardą. Konvertuotos DB rinkmenai nustatyti drausties požymį *Read-only* patariama ne tuoj pat, bet po tyrinėjimų, nes priešingu atveju nebus vykdomi visi teisėti veiksmai.

1.02 UŽDUOTIS.

1. Sukurti kitą duomenų bazės šablono *Expenses* (žr. PRATIMAS), pavadinti kitu vardu *DB_SANAUDOS*.
2. Padaryti atsargai DB šablono kopiją (žr.: 1.3 skyr. pradžia).
3. Iš 1.01 UŽDUOTIES jau žinant įrašų pavyzdžius užpildyti šablono *DB_SANAUDOS* lentelės savos redakcijos lietuviškais laukų vardais (langas *Design*) ir įrašais (langas *Datasheet*).
4. Tyrinėti nuosekliai savo suredaguotą DB, kaip ankstesnėje užduotyje.

Pastaba: pradedantieji 3 ir 4 punkto gali nevykdyti.

1.2. Kaip dirba specialiosios paskirties programa *Database Wizard*

PRATIMAS.

Prieš pradėdami tyrinėti naujai sukurta ir sutvarkyta duomenų bazę (DB) rinkmenai nustatykite požymį *Read-only*, tik skaityti, kad atsitiktinai nepakeistumėte anksčiau nustatytų duomenų. (*My Computer, C:\My Documents\Macro&VBA*, ties DB rinkmenos vardu spustelėkite pelės dešinįjį klavišą, bakstelėkite *Properties*, įjunkite jungiklį *Read-only* ir *Apply, Close*.) Dabar to daryti dar nereikia, nes sukurtas tik DB šablonas, ir negalėsite matyti visų rodinių. Jeigu nustatėte rinkmenos požymį *Read-only*, atsisakykite. Jeigu nenustatėte, pasimokykite – nustatykite, įkraukite DB, pradėkite apžvalgą ir atsisakykite.

Gražinkite į ekraną pagrindinį duomenų bazės langą: mygtukų juostoje *Form View*, esančioje virš lango *Main Switchboard*, bakstelėkite mygtuką *Database Window* arba bakstelėkite piktogramos *DB_EXPENSES:Database*, esančios lango *Main Switchboard* apačioje, kairėje, mygtuką *Restore* arba *Maximize*. Iš sutapdinto *Access* ir duomenų bazės lango *Microsoft Access - [DB_EXPENSES:Database]* į komutacinio skydelio langą *Main Switchboard* galėsite grįžti, kai bakstelėsite duomenų bazės lango mygtuką *Minimize Window* arba, kai pagrindinių objektų kortelėje *Forms* pažymėsite formą *Switchboard* ir bakstelėsite mygtuką *Open*.

Kai reikia dažnai kaitalioti langus iš komandų sąrašo *Window* nurodykite komandą *Cascade*. Sistemos lange *Microsoft Access* matysite abu langus *DB_EXPENSES:Database* ir *Main Switchboard*. Bakstelėdami giliau esančio lango išsikišusioje dalyje jį aktyvinsite. Taip galėsite greitai peržiūrėti tai vieną, tai kitą langą. Arba abu langus matysite ekrane, jei nurodysite, pvz.: *Window → Tile Vertically*.

Dabar pasirinkite patogų langų valdymo būdą, kad galėtumėte greitai pamatyti skirtingus langus, ir atkreipkite dėmesį į tokius momentus:

- Duomenų bazės lange, objektų kortelėje *Forms*, matome tik vieną formą *Switchboard*, kuri, kaip jau žinome, gali pateikti tris papildomus skydelius: *Main Switchboard*, *Forms Switchboard*, *Reports Switchboard* su skirtingos paskirties komandiniais mygtukais *Expenses*. Jeigu duomenų bazės lange atversite formos *Switchboard* konstravimo langą - *Switchboard:Form* (mygtukas *Design*), matysite 8 komandų mygtukus, kurie neturi pavadinimų. Galima spėti, nors tai ir nėra akivaizdu, kad programa *Database Wizard* panaudoja šią formą visiems komutaciniams skydeliams operatyviai parengti, kiekvienam išskirdama tinkamą komandos mygtuką. Pavyzdžiui, išskirdama *Switchboard:Form* antrąjį mygtuką pertvarko buvusią *Main Switchboard* į *Forms Switchboard*, o išskirdama trečiąjį mygtuką – į *Reports Switchboard*. *Database Wizard*, be abejo, sukuria komandas pavadinimams, komandų mygtukams ir komandų veiksmams atitinkamai keisti.
- Duomenų bazės objektų kortelėje *Tables* be kitų keturių duomenų lentelių yra lentelė *Switchboard Items*, komutacinio skydelio dėmenys/elementai. Peržiūrėkite lentelės struktūrą: *Tables → Switchboard Items → Design*. Lange *Switchboard Items:Table* matysite 5 duomenų laukus ir duomenų tipus, žemiau – srityje *Field Properties*, kortelėje *General*, vaizduojamos laukų savybės. Uždarykite langą, *Close*. Įkelkite duomenų įrašų lentelę - bakstelėkite mygtuką *Open*. Lentelės lauke *SwitchboardID* komutacinių skydelių identifikatoriai, lauke *ItemNumber* – dėmenų eilės numeriai, lauke *ItemText* – mygtukų pavadinimai. Lentelėje pateikiama visa informacija komutaciniams skydeliams sudaryti ir komandų veiksmams nurodyti. Be to, lentelė yra šaltinis formai *Switchboard* parengti.

- Duomenų bazėje nefiksuojamos jokios užklausos *Queries*, vadinasi, taikomojoje programoje *Expenses* saugomų užklausų nėra. Pagalbinėse meistrų programose *Form Wizard* ir *Report Wizard* vietoj saugomų užklausų formų ir ataskaitų įrašų šaltiniais yra *SQL* operatoriai. Daugelis *Database Wizard* parengiamų taikomųjų programų neturi saugomų užklausų ir charakteristikoms *RecordSource* bei *RowSource* tiesiogiai panaudojama *SQL* (*Structured Query Language*, struktūrinė užklausų kalba).
- Vienodo formato specializuota dialoginė forma *Report Date Range* yra panaudojama du kartus. *Database Wizard* sukuria komandas pavadinimams pakeisti priklausomai nuo pasirinkto *Reports Switchboard* komandos mygtuko.

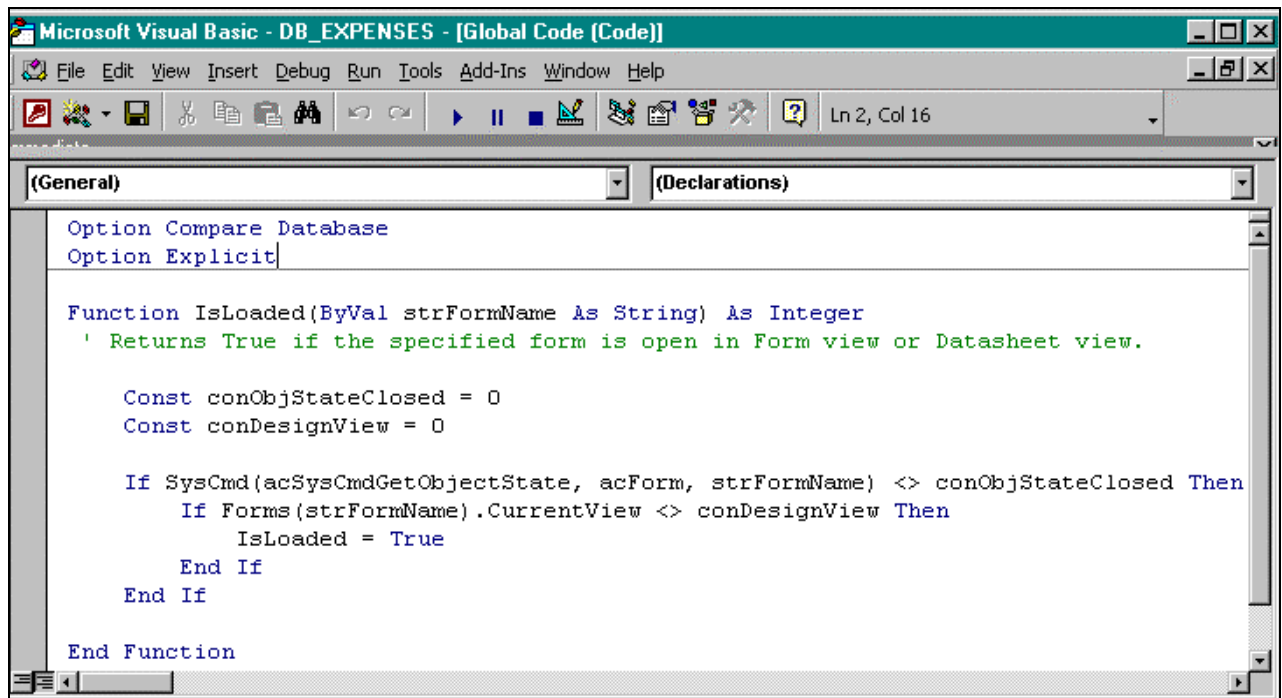
Langas Main Switchboard → Preview Reports → langas Reports Switchboard → pirmas komandos mygtukas Preview the Expense Rpt Summary by Category Reports → forma su laukais Beginning Rpt Date ir Ending Rpt Date ir mygtuku Preview, arba langas Reports Switchboard → antras komandos mygtukas Preview the Expense Rpt Summary by Employee Reports → to paties formato forma, t.y. su laukais Beginning Rpt Date ir Ending Rpt Date ir mygtuku Preview. Nagrinėjamos formos pavadinimai skiriasi tik paskutiniu žodžiu Category „Employee Kita apžvalga: duomenų bazės langas DB_EXPENSES:Database → Tables → Switchboard Items → Open → įrašų grupė 3 → laukų reikšmės.

- Makrokomandos pagrindinių objektų *Macros* kortelėje nevaizduojamos. Duomenų bazės langas DB_EXPENSES:Database → kortelė *Macros*. Modulių kortelėje *Modules* pateikiamas vienas standartinis modulis *Global Code*.

Panagrinėkime standartinį modulį *Global Code*. Duomenų bazės langas DB_EXPENSES:Database → kortelė *Modules* → pažymėkite *Global Code* → *Design*. Modulio lango *Global Code:Module* viršuje matysite dviejų išskleidžiamųjų sąrašų laukus (*General*) ir (*Declarations*) ir lango darbo srityje teksto dvi eilutes: *Option Compare Database*, *Option Explicit*. Lango apačioje, horizontaliosios slankiuotės eilutėje, kairėje pastebėkite du mygtukus.

Procedure View, procedūros vaizdas, *Full Module View*, programos vaizdas. Išbandykite jų veikimą.

Nustatytą pagal nutylėjimą nurodymą - išvesti kartu su procedūros vaizdu ir programos vaizdą - galima pakeisti: *Tools* → *Options* → kortelė *Editor*, išjungti jungiklį *Default to full Module View*. (Arba sistemai Access 97: *Tools* → *Options* → kortelė *Module* → skirsnis *Window Settings* → išjunkite jungiklį *Full Module View* → OK). Tada lango darbo srityje matysite tik *Procedure View* vaizdavimą. Jeigu pakeitėte, grąžinkite buvusį ankstesnį nustatymą. *Declarations* sąraše nurodomi sistemai Access vardai ir konstantos, kintamieji ir funkcijos – visi dėmenys, kurie turi būti pavartoti šiame modulyje. Atverkite sąrašą *Declarations* ir nustatykite *IsLoaded*. Matysite modulio (procedūros) komandas, parašytas programavimo kalba *Visual Basic for Applications* (*VBA*).



1.04 pav. Kai duomenų bazės modulių lange *Modules* yra pažymėtas modulis *Global Code*, pagrindinė programa/kodas, bakstelint mygtukų juostoje mygtuką *Code* programos lange išvedama procedūra-funkcija *IsLoaded*, kuri nustato ar atverta konkreti forma

PROCEDŪROS.

VBA atskiros vykdomųjų komandų grupės (jungtinių komandų vienetai) vadinamai *procedūromis*. Ši kartą modulį sudaro viena procedūra *IsLoaded*, prasidedanti žodžiu *Function*. Procedūros pabaigą žymi žodžiai *End Function*. Programoje *Expenses* procedūra *IsLoaded* panaudojama tik patvirtinti ar atverta duotoji forma ir neturi jokios įtakos formai arba ataskaitai atverti arba jų elementams derinti.

1.3. Komandos mygtuko veikimo procedūros tyrimas

Tęsimė duomenų bazės DB_EXPENSE tyrinėjimus. Tarkime, norim savo darbo aplanke/kataloge išsaugoti nepakeistą tik ką sukurtos duomenų bazės egzempliorių, o tyrinėjimams, redagavimui, modifikavimui reikia turėti tos pačios DB pat kitą egzempliorių. Padarykime DB kopiją savo darbo kataloge. Atverkite savo darbo aplanką, pvz.: *C:\My Documents\Macro&VBA*, ties DB vardu spustelėkite pelės dešinįjį klavišą, pasirinkite *Copy*, po to tame pat aplanke panašiai nurodykite *Paste*. Matysite kitą rinkmeną – esamos DB kopiją *Copy of DB_EXPENSES.mdb*. Toliau dirbsime ne su DB kopija, bet su tikrąja DB, t.y. *DB_EXPENSE*.

PRATIMAS. SU DB KOMANDŲ MYGTUKAIS YRA SIEJAMOS PAPILDOMOS PROCEDŪROS. PATYRINĖKIME VIENĄ IŠ JŲ.

1. Duomenų bazės lange *DB_EXPENSES:Database* atverkite formą *Expense Reports by Employee*: kortelėje *Forms* pažymėkite formos vardą ir bakstelėkite mygtuką *Open*.
2. Nustatykite formos veikseną *Design*, konstruoti: mygtukų juostoje *Form View*, kairėje, išskleiskite mygtukų sąrašą *View* ir bakstelėkite *Design View*.
3. Formos konstravimo lange *Expense Reports by Employee:Form* pažymėkite/bakstelėkite vardinį mygtuką *Expense Report Form* ir mygtukų juostoje bakstelėkite mygtuką *Properties*, savybės, arba vardinį mygtuką bakstelėkite du kartus. Ekrane pamatysite vardinio mygtuko savybių dialogo langą – *Command Button:Expense Report*, kuriame yra 5 kortelės: *Format*, *formatas*; *Data*, duomenys; *Event*, įvykis; *Other*, kitos savybės; *All*, visos savybės.

Šį kartą tiriamasis dėmuo/valdomasis elementas yra formos vardinis mygtukas - *Expense Report Form*. Savybių sąrašė yra vardijamos visos jo charakteristikos, kurios buvo nustatytos projektuojant mygtuką.

Įvykiai – *events* – Savybių rinkinys, dabar aprašantis nagrinėjamąjį dėmenį (toliau sakysime - objektą), vadinamas *d ė m ė n s / o b j e k t o b ū s ė n a*. Pavyzdžiui, bakstelėjus mygtuką, jis pažymimas. Būsenos pakeitimas *VBA* aplinkoje turi savitą interpretavimą – yra vadinamas *į v y k i u*. Mygtuko bakstelėjimas pele ženklinamas įvykiu *Click*. Kiekvienam įvykiui atitinka tam tikra įvykio savybė/charakteristika. Daugeliu atvejų įvykių savybės/charakteristikos yra pradedamos angliškuoju priešdėliu *On*, po kurio rašomas įvykio pavadinimas, pavyzdžiui, įvykiui *Click*, įvykio savybė yra - *On Click*.

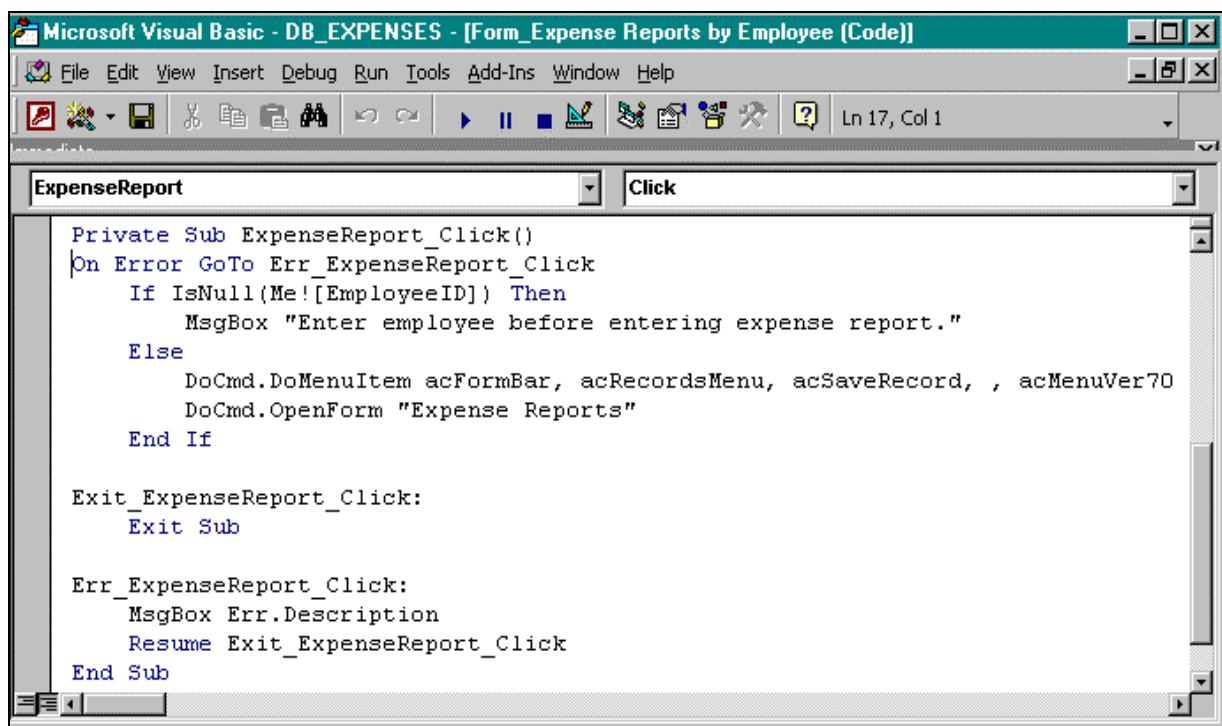
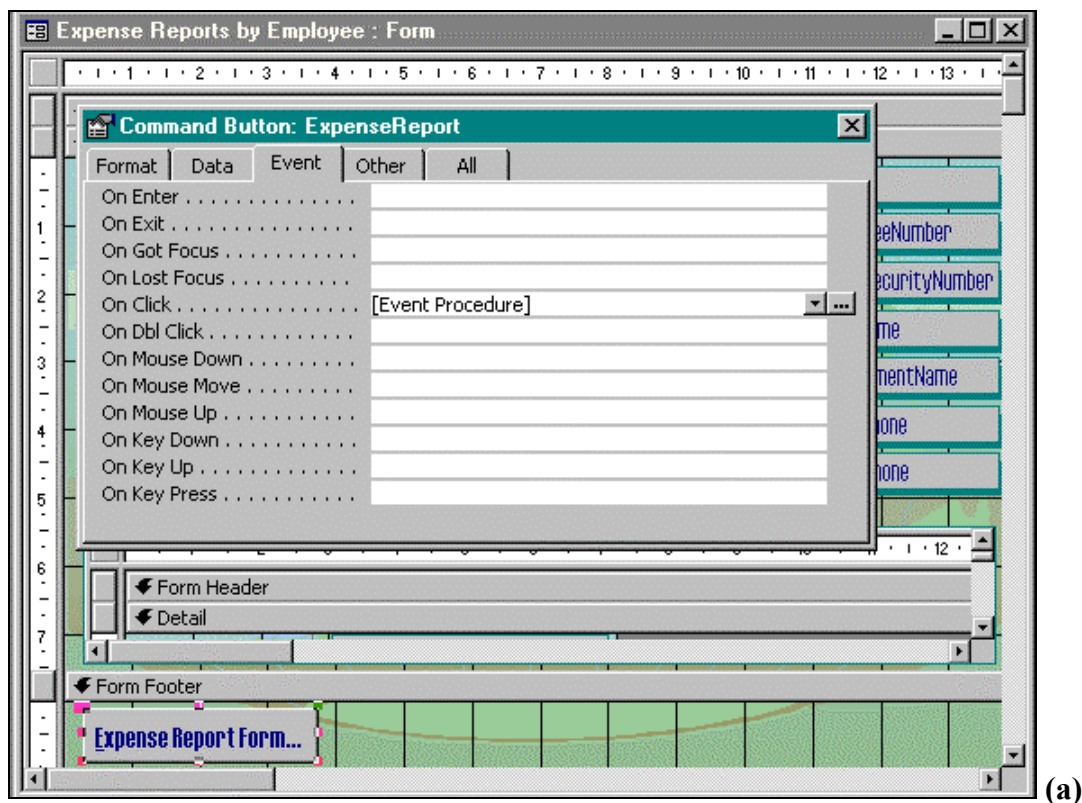
Objektas – *object* – bet kuri esmė ar realybė *VBA* aplinkoje (lentelė, laukas, mygtukas ar kitoks valdymo elementas ir t.t.), kurią galima apibūdinti. Gilindamiesi į programavimą labai dažnai vartosime žodį objektas.

4. Savybių dialogo lange *Expense Reports by Employee:Form* atverkite kortelę *Event*. Matysite įvykių savybių sąrašą. Į v y k i o s a v y b ė žymi prielaidą veiksmo, pakeisiančio dėmens būseną. Pavyzdžiui, įvykio savybė *On Mouse Down* žymi pelės žymeklio poslinkį ir mygtuko bakstelėjimo veiksmą, atpažįstantį ir pakeičiantį visus įvykius su vardu *On Mouse Down*. Išsamią informaciją apie įvykių savybes galima matyti žinyne *Help*. Atverkite komandų sąrašą *Help*, nurodykite *What is This?* ir bakstelkite prie žymikliu-klaustuku savybių sąrašė norimos savybės vardą. Galėsite paskaityti nurodytos savybės konsultacijos *Help* temą.

Įvykių panaudojimas labai išplečia programavimo galimybes. Programuojant rašomas komandų sąrašas (tai ir yra programa) ir po to pateikiamas sistemai *Access* jį vykdyti, esant tam tikram dėmens įvykiui. Toks programavimas vadinamas *į v y k i a i s o r i e n t u o t a s a r b a į v y k i a i s v a l d o m a s p r o g r a m a v i m a s* (*event-driven programming*).

Dialogo lange *Expense Reports by Employee:Form* įvykio savybės *On Click* reikšmė yra [*Event Procedure*]. Tai reiškia, kad yra parengta *VBA* procedūra, kuri skirta veikti pagal vardinio komandos mygtuko - *Expense Report Form* įvykį *Click*. Sistema *Access* vykdo tą procedūrą, kai bakstelimas komandos mygtukas. *VBA* procedūra, sužadinama objektui atpažinus įvykį, vadinama *į v y k i o p r o c e d ū r a* arba *į v y k i o a p d o r o j i m o p r o c e d ū r a*. Procedūros turinį pamatysite ekrane, kai eilutės *On Click* dešinėje bakstelėsite mygtuką *Build*, pažymėtą trim taškais. Pasidomėkite lango pavadinimu *Form Expense Reports by Employee:Class Module*, t.y. dabar matote formos *Expense Reports by Employee* modulį.

Sistema *Access* automatiškai vardija modulius pagal formą, panaudodama jos vardą su priešdėliu *Form_*. Bendra sintaksė: *Form_formos-vardas*. Panašiai vardijami moduliai pagal ataskaitų pavadinimus: *Report_ataskaitos-vardas*.



1.05 pav. Kai formos konstravimo lange pažymėtas komandos mygtukas *Expense Report Form*, bakstelint mygtukų juostoje mygtuką *Properties* išvedamas savybių sąrašas *Command Button: ExpenseReport*; sąrašė įvykių *Event* kortelėje bakstelint laukelyje *On Click*, o po to laukelio dešinėje atsiradusį mygtuką su trim taškeliais (a) išvedama įvykio savybės *Click* apdorojimo procedūra *ExpenseReport_Click* (b)

Pastaba: procedūrose sakinius *MsgBox* ir kt. galima redaguoti, pvz., parašyti lietuviškai.

Peržiūrėkite įvykių sąrašą ir įvykio *Click*, procedūros *ExpenseReport_Click* turinį. Po to peržiūrėkite formos lange pagrindinės ir pagalbinės formos laukų sąrašus ir nurodydami *View* →

Field List. Prieš nurodant komandą liniuočių sankirtose reikia įjungti bevardžius stačiakampius mygtukus.

Programa *Database Wizard* sukuria procedūras kiekvienam komandos mygtukui, taip automatizuojamas jų veikimas. Mokant programuoti galima visiškai automatizuoti duomenų bazės valdymą.

5. Uždarykite visus langus ir grįžkite į duomenų bazės langą. Baikite darbą su *Access* arba tęskite tyrinėjimus. Susipažinkite su kitomis *Access* priemonėmis.

1.03 UŽDUOTIS

1. Kartojant DB šablono *Expenses* automatizuoto projektavimo veiksmų seką (1 skyrius, PRATIMAS) pasirinkti ir savarankiškai parengti kitos DB šablono.
2. Sekant 1.1 – 1.3 skyreliuose aprašytą DB šablono *Expenses* nagrinėjimą panašiai tyrinėti naujai suprojektuotą DB šablono, pvz., *Contact Management*. Sulietuvinti laukus ir reikšmes. Pastebėti ir įsidėmėti ypatumus.
3. Padaryti atsarginę DB šablono kopiją.
4. Užpildyti naujos DB (ne kopijos) lentelės savais laukų vardais ir 3 – 5 įrašais.
5. Patikrinti DB veikimą.
6. DB šablono projektavimo ir DB trumpą aiškinimą reikia pateikti atskira Word rinkmena ir įrašyti savo darbo aplanke.

Pastaba: pradedantieji šios užduoties gali nevykdyti.

1.4. Paslaugų programos *Command Button Wizard* panaudojimas

Nagrinėdami komutacinio skydelio *Main Switchboard* mygtukais *Expenses* vykdomas komandas jau pastebėjome vienkryptę eigą.

Prisiminkime: bakstelėkite pirmąjį mygtuką - *Enter/View Expense Reports by Employee*, po to formos vardinį mygtuką *Expense Report Form*, po to kitoje formoje vardinį mygtuką *Preview Reports* - pamatysite ataskaitos lapą *Expense Report*. Tai ir yra vienkryptis maršrutas – nuo duomenų banko pagrindinio komutacinio skydelio iki parengtos ataskaitos. Dabar norėdami sugrįžti atgal turėsite nuosekliai uždaryti visus atvertus langus.

1 PRATIMAS.

Paslaugų programa *Command Button Wizard* leidžia parengti dvikryptį maršrutą. Jį gali automatiškai parengti atitinkamą grįžtamojo judėjimo komandos mygtuko ir įvykio *Click* apdorojimo procedūrą *VBA* kalba.

1. Duomenų banko lange *Microsoft Access* - [DB_EXPENSES:Database] atverkite formą *Expense Reports by Employee* bakstelėdami mygtuką *Design*, konstruoti.
2. Jeigu ekrane nėra konstravimo mygtukų skydelio – *Toolbox*, mygtukų juostoje bakstelėkite mygtuką *Toolbox*. Jeigu skydelyje *Toolbox* neišspaus tas mygtukas *Control Wizards*, valdymo vedliai, bakstelėkite/įspauskite jį.
3. *Toolbox* mygtukų skydelyje bakstelėkite/įspauskite mygtuką *Command Button*, komandų mygtukas, o po to bakstelėkite formoje, dešiniau vardinio mygtuko *Expense Report Form*. Bakstelėtoje vietoje atsiras žymėjimo taškais apsuptas vardinis mygtukas, kurio pradinį vardą sistema užrašo pati, pvz., *Command30*, arba vietoj 30 gali būti kitoks vidinis numeris. Todėl aiškinimuose vietoj *Command30* bus rašoma *CommandX*. Čia *X* - sistemos *Access* parenkamas valdomojo elemento - dabar mygtuko vidinis numeris.

Ekране, aukščiau, pamatysite pirmąjį meistro dialogo langą *Command Button Wizard*, kuriame informuojama: *What action do you want to happen when the button is pressed?* (Kokio lauksite veiksmo, kai spustelėsite mygtuką?) *Different actions are available for each category* (veiksmai priklauso nuo užduočių kategorijos).

Peržiūrėdami sąrašus *Categories* ir *Actions* galime suprasti, kad paslaugų programa meistras *Command Button Wizard* gali parengti procedūras daugeliui tipinių operacijų.

Paslaugų programa *Command Button Wizard* automatizuojamos operacijos

2 lentelė

Eil. Nr.	Categories Užduočių kategorijos	Actions Judėjimai, veiksmai
1	<i>Record Navigation</i> , žymiklio perkėlimas	<i>Go to First Record</i> , į pirmąjį įrašą; <i>Go to Last</i> , į paskutinįjį; <i>Go to Next</i> , į kitą; <i>Go to Previous</i> , į prieš esantį; <i>Find Record</i> , rasti įrašą; <i>Find Next</i> , rasti kitą.
2	<i>Record Operations</i> , įrašų operacijos	<i>Add New Record</i> , papildyti nauju įrašu; <i>Delete Record</i> , šalinti įrašą; <i>Duplicate Record</i> , dubliuoti įrašą; <i>Print Record</i> , spausdinti įrašą; <i>Save Record</i> , išsaugoti įrašą; <i>Undo Record</i> , atšaukti operaciją, siejama su įrašu.
3	<i>Form Operations</i> , formų operacijos	<i>Apply Form Filter</i> , panaudoti filtrą; <i>Edit Form Filter</i> , redaguoti filtrą; <i>Close Form</i> , uždaryti formą; <i>Open Form</i> , atverti formą; <i>Print a Form</i> , spausdinti formą; <i>Print Current Form</i> , spausdinti dabartinę formą; <i>Refresh Form Data</i> , atnaujinti formos duomenis.
4	<i>Report Operations</i> , ataskaitų operacijos	<i>Print Report</i> , spausdinti ataskaitą; <i>Preview Report</i> , peržiūrėti ataskaitą; <i>Mail Report</i> , siųsti ataskaitą paštu; <i>Send Report to File</i> , išvesti ataskaitą į rinkmeną/failą.
5	<i>Application</i> , taikomoji programa	<i>Run Notepad</i> , paleisti <i>Notepad</i> ; <i>Run MS Word</i> , paleisti <i>MS Word</i> ; <i>Run MS Excel</i> , paleisti <i>MS Excel</i> ; <i>Quit Application</i> , baigti darbą su programa; <i>Run Application</i> , paleisti programą.
6	<i>Miscellaneous</i> , Įvairios	<i>Print Table</i> , spausdinti lentelę; <i>Run Macro</i> , vykdyti makrokomandą; <i>Run Query</i> , vykdyti užklausą; <i>AutoDialer</i> , rinkti automatiškai numerį.

2 PRATIMAS.

Kaip konstruoti mygtuką uždaryti formai *Expense Reports by Employee* ir grįžti į pagrindinį komutacinį skydelį *Main Switchboard*:

1. *Command Button Wizard* dialogo pirmajame lange nurodykite kategoriją *Categories* sąrašė *Form Operations*, formų operacijos, ir veiksmą *Actions* – *Close Form*, uždaryti formą. Bakstelėkite *Next*.

2. Kitame *Command Button Wizard* lange galima projektuoti mygtuko išvaizdą. Klausima: *Do you want text or a picture or the button?* (Norite matyti ant mygtuko tekstą ar paveikslėlį?) *If you choose Text, you can type the text to display. If you choose Picture, you can click Browse to find a picture to display* (Jei pasirinksite *Text*, galėsite parašyti tekstą. Jei pasirinksite *Picture*, galėsite bakstelėti *Browse* ir suieškoti paveikslėlį).

Ijunkite jungiklį *Text*, turi būti pažymėtas *Exit*. Bakstelėkite *Next*. (*Access 97*: teksto įvesties lauke surinkite anglišką žodį *Return*, grįžtis).

3. Trečiame *Command Button Wizard* lange paskaitykite informaciją: *What do you want to name the button?* (Kaip įvardysite mygtuką?) *A meaningful name will help you to refer to the button later* (prasmingas vardas leis greičiau pastebėti mygtuką vėliau).

Įvesties lauke surinkite mygtuko vardą *cmdReturn*. Žemiau pranešama, kad vedlys jau surinko visas reikiamas žinias – *That's all the information the wizard needs to create your command button*. Lango apačioje yra jungiklis *Display Help on customizing the button*, kurį įjungus galima būtų matyti aiškinimus. Šį kartą jo įjungti nereikia.

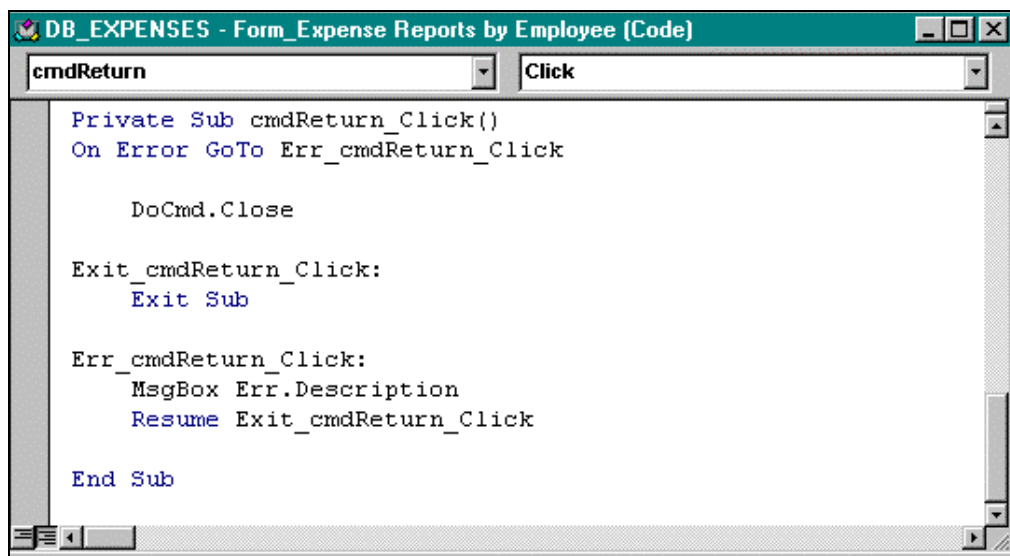
4. Kai dabar bakstelėsite lango mygtuką *Finish*, meistras *Wizard* parengs komandos mygtuką ir pateiks *VAB* procedūrą nurodytam veiksmui atlikti.

5. Analizuokime procedūrą: kai naujas komandos mygtukas *Close Form* pažymėtas, mygtukų juostoje bakstelėkite mygtuką *Properties*, savybės. Mygtuko savybių lange *Command Button:cmdReturn*, kai atverta kortelė *Event*, įvykis, bakstelėkite savybės *On Click* įvesties eilutės dešinėje, už savybės reikšmės [*Event Procedure*], pastebėsite mygtuką su trim taškeliais, tada bakstelėkite jį.

Kai procedūrų lango *Form_Expense Reports by Employee (Code)* išskleidžiamajame sąrašė, viršuje, kairėje, nustatytas mygtuko vardas *cmdReturn*, o sąrašė, dešinėje, įvykis *Click*, matysite programavimo kalba *VBA* parengtą formos uždarymo procedūros *cmdReturn_Click* tekstą (žr. žemiau 1.06 pav.).

6. Uždarykite procedūros demonstravimo langą, mygtuko savybių langą, atsisakykite mygtuko *Close Form* žymėjimo. Dabar formoje du vardiniai mygtukai: *Expense Report Form* ir *Close Form* (*Access 97* vietoj *Close Form* mygtukas *Return*). Mygtukų juostoje bakstelėkite mygtuką *Save* įrašyti pakeitimą.

7. Nustatykite formos vaizdavimo rodinį - bakstelėkite mygtuką *Form View*. Išbandykite mygtuką *Expense Report Form* ir mygtuką *Close Form*. Jeigu mygtukas neveiks, reikia redaguoti procedūrą, vėl nustatykite formos konstravimo veiksena *Design View*, du kartus bakstelėkite mygtuką, savybių lange, eilutėje *On Click*, dešinėje, bakstelėkite mygtuką *Build* (su trim taškais) ir procedūros lange redaguokite trečiąją eilutę, esančią žemiau *Private Sub... ir On Error GoTo...* Trečioje eilutėje turi būti tik komanda formai uždaryti: *DoCmd.Close*. Redaguokite eilutę, įrašykite ir vėl išbandykite vardinių mygtukų veikimą. Mygtukai veikia tik formos vaizdavimo lange *Form View*. Įkelkite langą *Main Switchboard* ir bakstelėkite pirmąjį mygtuką, o formas vaizdavimo lange mygtuką *Close Form*. Dabar sistema *Access* turi vėl grąžinti *Main Switchboard* langą. Jeigu taip - puiku! Įsidėmėjote, kad esant būtinumui, procedūrą galima redaguoti.



1.06 pav. Sistemos *Access* programavimo kalba *VBA* automatiškai parengta mygtuko *Close Form* įvykiui *Click* apdoroti procedūra *cmdReturn_Click*, uždaranti formą *Expense Reports by Employee*

1.04 UŽDUOTIS.

1. Ataskaitoje *Expense Report* sukonstruoti ataskaitos uždarymo komandos mygtuką.
2. Savo savarankiškai pasirinktoje automatizuotai sukurtoje duomenų bazėje pasirinkti formą arba ataskaita ir įtaisyti formos arba ataskaitos uždarymo komandos mygtuką.

Pastaba: pradedantieji užduoties antro punkto gali nevykdyti.

1.5. Hipersaitai ir nuorodos

H i p e r s a i t a i - tai hipertekstų programos priemonė, sudaranti galimybę greitai persikelti į kitą vietą: į kitą vietą lange, į kitą langą, rinkmeną ir t. t. **H i p e r s a i t o n u o r o d a** tai išskirta teksto dalis, piešinys, vaizdo dalis, komanda komandų sąrašė ar komandos mygtukas. Vartotojas pele bakstelėdamas hipernuorodą gali persikelti į kitą duomenų bazės objektą arba į kitą duomenų bazę, į sistemomis *Word*, *Excel* ar *PowerPoint* parengtą rinkmeną/dokumentą, taip pat į rinkmeną, pasiekiamą per vietinį kompiuterių tinklą arba interneto tinklo *WWW* puslapį.

WWW žiniatinklui nurodyti adresus rašomas taikant standarto *URL – Uniform Resource Locator* – universalusis interneto išteklių adresas, reikalavimus, pvz.: <http://www.microsoft.com/>; <http://www.urm.lt/about/>

Katalogui arba rinkmenai nurodyti kompiuterio failų sistemoje taikomas universalusis kelio/maršruto adresavimas *UNC – universal naming convention*, pvz.: *C:\Program Files\Microsoft Office\Access\Samples\Nwind.mdb*.

Hipertekstas – *hypertext* – ekrane matomas tekstas, kurio kai kurie terminai paryškinti, jų apibrėžtis galima greitai pamatyti, pvz., bakstelint pele.

Hipersaitas – *hyperlink* – hiperteksto saitas (*link*) su kitu tekstu ar dokumentu, hipersaito nuoroda paryškinama ir pabraukiama, ją bakstelint greitai persikeliama į nurodytą vietą.

Sistema *Access* hipersaitus gali saugoti atskirai duomenų lentelėje arba formos susietame valdomajame elemente, kuris turi jungtį su hipersaito *Hyperlink* lauku.

SĄRYŠINĖS HIPERNUORODOS DUOMENŲ LENTELĖJE.

Access 97 ir *Access 2000* yra numatytas duomenų tipas *Hyperlink*, hipersaitas/hiperjungtis, kuris tinka hipernuorodų adresams saugoti. Pavyzdžiui, interneto tinklo paslaugų tiekėjų adresai saugomi atskirame lauke, panašiai kaip kita kontaktų informacija ir pagal reikalą perkeliama į tiekėjų duomenų formą.

1 PRATIMAS. DUOMENŲ BANKO *DB_EXPENSES* LENTELĘ *EMPLOYEES*, *TARNAUTOJAI*, PAPILDYKITE LAUKU, KURIO DUOMENŲ TIPAS *HYPERLINK*. LAUKAS SKIRTAS SAUGOTI *URL* ADRESAMS.

1. Duomenų banko lange *DB_EXPENSES:Database* atverkite lentelę *Employees*. Nustatykite konstravimo *Design* veikseną. *Fild Name* stulpelyje, žemiau lauko *Notes*, klaviatūra surinkite naujo lauko vardą *HomePage*. Stulpelyje *Data Type* iš sąrašo nurodykite duomenų tipą *Hyperlink*. Srities *Field Properties* laukelyje *Caption* įrašykite *Home Page*.

2. Įrašykite lentelę – bakstelėkite mygtuką *Save* ir atverkite įrašų lentelę *Datasheet* – bakstelėkite mygtuką *Datasheet View*.

3. Hipersaito nuorodoms įvesti į įrašų lentelę pasirinkime paprasčiausią būdą - mygtukų juostoje esantį mygtuką *Insert Hyperlink*, įterpti hipersaitą.

Nustatykite įvesties žymeklį lauko *Home Page* pirmoje ląstelėje. Reikia surinkti *Nacy's Page*. *Nancy* - įstaigoje dirbančios darbuotojos vardas (žr. lauką *First Name*). Galite kopijuoti *Nancy* iš lauko *First Name*, o likusią teksto dalį surinkti klaviatūra. Į teksto pakitusią spalvą ir pabraukimą dabar nekreipkite dėmesio. Mygtukų juostoje bakstelėkite mygtuką *Insert Hyperlink*. Ekrane pamatysite hipersaitų redagavimo langą *Edit Hyperlink*.

Hipersaito adresą galima įvesti lauke *Type the file or Web page name*.

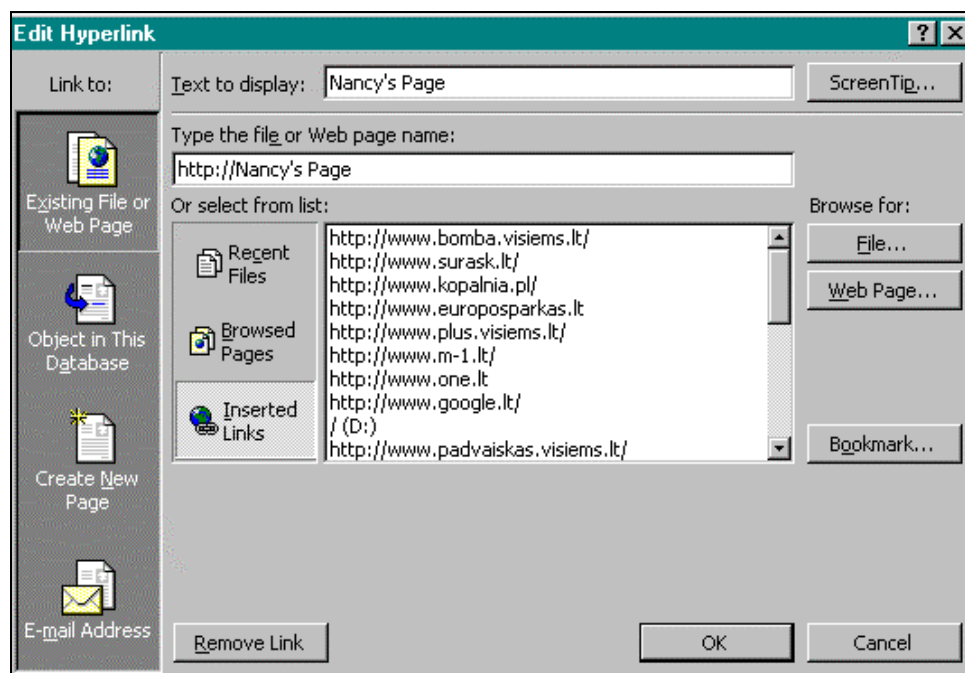
(*Access 97* laukas vadinamas kitaip: *Link to file or URL*). Be to, *Access 97* visas dialogas kitoks. Čia toliau pateikiamas aprašas iki pastraipos pabaigos tinka tik *Access 97*: Paskaitykite žemiau esančią informaciją: *Enter or locate the path to the document you want to link to. This can be an Internet address (URL), a document on your hard drive, or a document on your company's network* (surinkite arba nurodykite kelią iki reikiamo dokumento: interneto adresą (*URL*) ar dokumento, esančio kompiuteryje arba pasiekiamo vidiniu tinklu). Lauke *Named location in file (optional)* galima nurodyti papildomą/vidinį adresą. Apie tai informuojama žemiau: *If you want to jump to a specific location within the document, such as a bookmark, a named range, a database object, or locate that information above* (jei reikia staiga persikelti į dokumento vietą, paženklinimą gairele, vietos žymeniu, skaidrės numeriu arba duomenų bazės objekto vardu įveskite papildomą informaciją).

Bendruoju atveju *Hyperlink* lauko reikšmę gali sudaryti trys dalys:

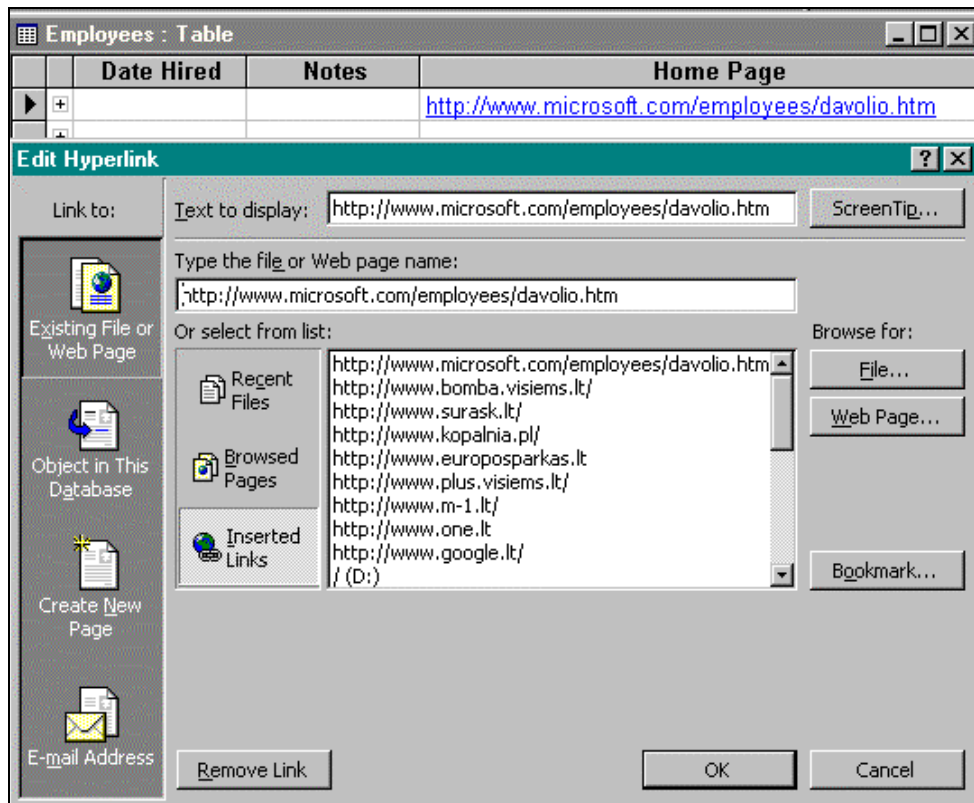
tekstas#adresas# papildomas_adresas#

4. Pirmoji dalis „tekstas“ nebūtina. Pašalinkite tekstą iš lauko *Type the file or Web page name* (1.07 pav.), (*Access 97* laukas kitu vardu: *Link to file or URL*) ir surinkite tokį adresą:

<http://www.microsoft.com/employees/davolio.htm>



1.07 pav. *Edit Hyperlink* dialogo langas hipersaito adresui redaguoti ir įvesti



1.08 pav. Hipersaito adreso nurodymas iš sąrašo *Or select from list* dialogo lange *Edit Hyperlink*

Tai pirmosios tarnautojos fiktyvus žiniatinklio adresas. Bakstelėkite *OK*. Jeigu įrašų lentelės lauke *Home Page* vis dar matote ankstesnį tekstą *Nancy's Page*, pašalinkite jį. Vėl bakstelėkite mygtuką *Insert Hyperlink* ir dar kartą surinkite fiktyvų adresą, kurį dabar galima nurodyti iš sąrašo *Or select from list* (1.08 pav.) ir bakstelkite *OK*. Dabar įrašų lentelėje *Employees* lauko *Home Page* pirmoje ląstelėje matysite nebe tekstą, bet *URL* adresą (1.08 pav.).

Įsidėmėkite: nors tekstas, rašomas prieš adresą, nėra būtinas, tačiau jeigu jis surinktas, tai ir lauko ląstelėje rodomas tik tekstas, bet ne adresas. Jeigu teksto nėra, rodomas adresas.

5. Įvestas adresas jau bus įrašytas, kai įrašų lentelėje įvesties žymiklį nukelsite į žemiau esančią ląstelę. Jeigu adresas būtų ne fiktyvus, bet realus, reikėtų tik bakstelėti adreso ląstelėje (pelės rodyklė tampa ranka su ištiestu rodomuoju pirštu) ir naršyklė *Internet Explorer* atverstų nurodytą dokumentą.

6. Jeigu norėsite hipersaito nuorodą (adresą) keisti ar redaguoti, spustelėkite pelės dešinįjį klavišą, kai rodyklė adreso ląstelėje, ir iš parankinio komandų sąrašo nurodykite *Hyperlink → Edit Hyperlink*. Lauke *Text to display* galėsite redaguoti hipersaito tekstą. Pavyzdžiui, surinkite vėl buvusį tekstą *Nancy's Page*. Įrašų lentelės lauke *Home Page* vėl matysite tekstą. Redaguokite, ištrinkite tekstą. Matysite ankščiau surinktą adresą.

7. Duomenų bazės lange atverskite formos *Expense Reports by Employee* langą (*Open*). Formos lange naujo lauko *Home Page* nėra. Nustatykite konstravimo režimą *Design View*. Iš komandų sąrašo *View* nurodykite *Field List*. Iš laukų sąrašo *Employees* tempkite lauką *Home Page* ir palikite žemiau paskutinio lauko *Work Phone* (1.09 pav.).

1.09 pav. Siejamo hipersaito *Home Page* rodinys formoje *Expense Reports by Employee*

Jeigu reikia padaryti naujam laukui vietos, pagalbinės formos sritį *Employees Subform* tempkite žemyn. Bakstelėkite baltame fone, po to, kai pelės rodyklę nustatysite ties srities linija ir ji įgaus rankos išvaizdą, galėsite tempti pagalbinės formos sritį žemyn. Lygiuokite lauką *Home Page* pagal kitus laukus stulpelyje. Pažymėkite, pasimokykite nutempti į kitą vietą, kai pelės rodyklė vaizduojama plaštaka arba rankos rodomuoju pirštu.

8. Peržiūrėkite formą vaizdavimo lange *Form Wiew*. Po to redaguokite, kad būtų galima matyti visą adresą. Konstravimo lange išplėskite adreso lauką ir vėl peržiūrėkite formą. Kai bakstelėsite mygtuką *Close Form*, ekrane pasirodys pranešimas: *Do you want to save changes to the design of form 'Expense Reports by Employee'?* (Ar įrašyti pakeitimus?). Bakstelėkite *Yes*.

Kitus įrašus (*Expenses* atveju 2 ir 3 įrašą) lauke *Home Page* galima tai pat papildyti savais adresais. Taip ir darykite. Dabar *Edit Hyperlink* lange adresus nurodykite iš sąrašo *Or select from list*.

Susipažinome, kaip gali būti saugomos su valdomuoju elementu – lauku *Home Page* siejamos hipersaitų nuorodos.

1.05 UŽDUOTIS.

1. Duomenų bazės DB_SANAUDOS, kurioje įtaisytas objekto uždarymo komandos mygtukas įterpti lauką su *URL* adresu.
2. Automatizuotai sukurtos savo duomenų bazės formoje arba ataskaitoje įterpti lauką su *URL* adresu.

Laisvos/nesiejamos hipersaitų nuorodos. Laisvos hipernuorodos, pvz. pereiti iš vienos formos į kitą arba iš formos į ataskaitą įtaisomos tiesiog pačioje formoje panaudojant žymenį (*label*), komandos mygtuką (*command button*) arba atvaizdą (*image*). Hipernuorodos aprašui parengti patogų panaudoti savybę *Caption* (pagal žymenį ir komandos mygtuką). Hipernuorodai rengti nustatomos valdomojo elemento savybės *HyperlinkAddress* ir *HyperlinkSubAddress*. Rengiant hipernuorodą į tos pačios duomenų bazės kitą objektą savybė *HyperlinkSubAddress* nenurodoma (paliekamas tuščias laukas).

2 PRATIMAS. HIPERSAITO RENGIMAS PANAUDOJANT ŽYMENĮ *LABEL*

Laisvų hipersaitų tyrimui reikia įrengti hipersaitų valdomuosius elementus ir nustatyti jų savybes. Parenkite hipersaito valdomąjį elementą: a) panaudojant žymenį, kuris atveria kitą tos pačios duomenų bazės objektą; b) panaudojant atvaizdą, kurį bakstelėjus atveriamas kitos duomenų bazės objektas; c) panaudojant komandos mygtuką, aktyvinantį naršyklę, kuri įkelia *WWW* puslapį.

Valdomasis elementas – *control* – bet kuris dažniausiai elementarus duomenų formos struktūros dėmuo: mygtukas, jungiklis, lauko pavadinimas, reikšmės laukas, linija, apvadas ir kt., kuriuo arba inicijuojamas koks nors valdymo veiksmas, arba ir pats gali būti valdomas (redaguojamas, perkeliamas į kitą vietą, jungiamas į grupę, išlaptinamas ir t.t.)

1. Duomenų bazėje *DB_EXPENSES* sukurkite naują tuščią formą *frmHyperlinks* (DB langas, *Forms*, mygtukas *New*, dialogo lange *New Form* pasirinkti *Design View*, OK, mygtukas *Save*, langelio *Save As* laukelyje *Form Name* surinkti *frmHyperlinks*, OK). Konstravimo lange du kartus bakstelkite liniuotųjų sankirtoje stačiakampį mygtuką, atverkite formos savybių sąrašą, savybės *Caption* reikšmę surinkite *Hyperlinks*.

Valdomųjų elementų skydelyje *Toolbox* bakstelėkite mygtuką valdomojo elemento *Label* ir bakstelėkite formos darbo srityje, kairėje. Pažymėtoje vietoje surinkite hipersaito pavadinimą *Hyperlinks*. Po to pažymėkite įvestą pavadinimą ir mygtukų juostoje bakstelėkite mygtuką *Properties*.

Savybių dialogo lange *Label:LabelX* (čia X skaičius, jį parašo pati sistema, pvz., 10) raskite ir bakstelėkite savybę *Hyperlink SubAddress*, po to bakstelėkite įvesties lauko dešinėje atsiradusį mygtuką su trim taškeliais *Build*.

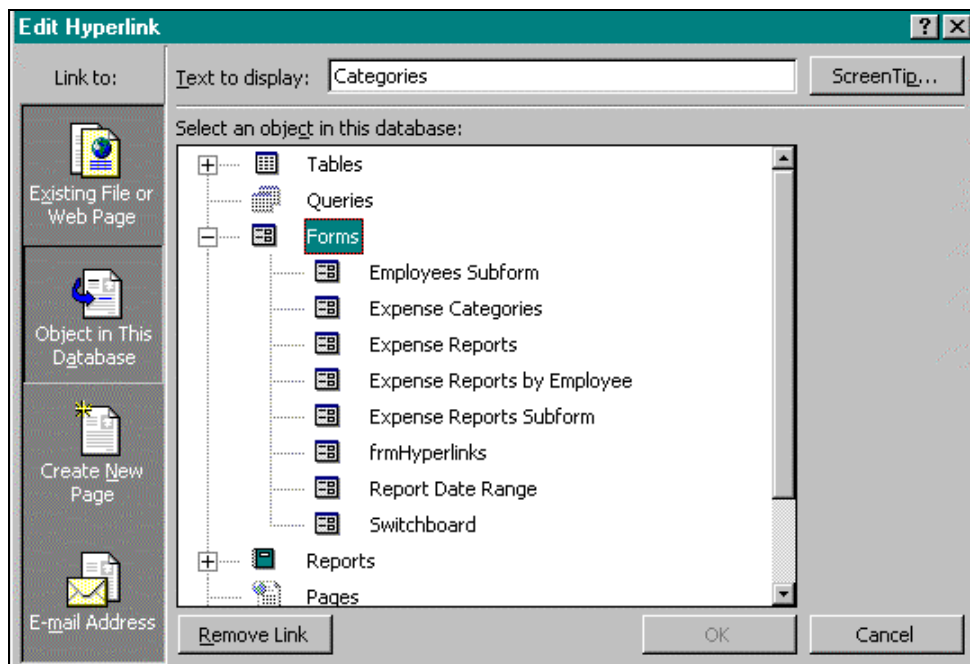
2. Dialogo lange *Insert Hyperlink* bakstelėkite mygtuką *Object in This Database*, galėsite parinkti ir nurodyti objektą - taikinį, pvz., formą *Expense Categories* (1.10 pav.). Pažymėkite *Expense Categories* ir bakstelėkite OK.

(*Access 97* pasirenkama kitaip: Įvesties lauką *Link to file or URL* palikite tuščią, taip pranešama sistemai, kad ieškomas objektas - taikinytis yra dabartinėje rinkmenoje, o ne tinkle ir ne kitame kataloge arba kitoje rinkmenoje. Bakstelėkite mygtuką *Browse*, esantį įvesties lauko *Named location in file* dešinėje. Į ekraną išvedamas dialogo langas *Select Location*, kuriame pateikti visi dabartinio duomenų bazės objektai. Lange *Select Location* atverskite kortelę *Forms*, pažymėkite reikiamą objektą *Expense Categories*. Kitu atveju (kai reikia) galima nurodyti kitą objektą. Bakstelėkite OK lange *Select Location*. Lango *Insert Hyperlink* lauke *Named Location...* dabar jau matote objekto - taikinio pavadinimą *Form Expense Categories*. Be to, dabar jau įjungtas jungiklis *Use relative path for hyperlink*, hipersaito kelias pagal nustatymą. Bakstelėkite OK.)

Dabar savybių sąrašo lange, eilutėje *Hyperlink SubAddress*, nustatyta reikšmė *Form Expense Categories*. Uždarykite savybių langą *Label:LabelX*. Sistema *Access* sukuria hipersaito valdomąjį elementą *Categories*.

4. Įrašykite formą – mygtukas *Save*. Atverkite formos vaizdavimo langą *Form View*. Bakstelėkite hipersaito valdomąjį elementą *Categories*. Ekrane matysite to paties duomenų banko kitą objektą – formą *Expense Categories*.

5. Uždarykite formos *Expense Categories* langą – mygtukas *Close*. Uždarykite formos *Expense Reports by Employee* langą – mygtukas *Close Form*. Grįšite į duomenų bazės langą *DB_EXPENSES: Database*.



1.10 pav. Dialogo lange *Insert Hyperlink* bakstelint stulpelyje *Link to* mygtuką *Object in This Database* išvedamas veikiamosios duomenų bazės pagrindinių objektų medis ir pasirenkamas hipersaito objektas.

1.06 UŽDUOTIS.

1. Duomenų bazėje DB_SĄNAUDOS arba savo automatizuotai suprojektuotoje DB sukurkite naują tuščią formą ir joje nustatykite laisvą/nesiejamą hipersaito nuorodą į DB kitą objektą - formą ar ataskaitą.
2. Iš savo darbo aplanko arba kompaktinio disko (*\Web\expenses.mdb*) įkraukite atsisiustą DB *Expenses*, (pvz.: *C:\Vadovas_VBA\AtsiustaInfo\expenses.mdb*), atverkite formą *frmHyperlinks*, išbandykite įterptų formoje hipersaitų valdomųjų elementų: *Categories*, paveikslėlio ir *Home Page* veikimą.

Pastaba: atsisiustą DB *Expenses* galėsite tik skaityti, nes rinkmenai nustatytas požymis „tik skaityti“, be to, yra numatytas slaptažodis DB įkrauti ir nustatytas kai kurių objektų bei parametrų vaizdavimo ribojimas (kaip nustatyti ribojimą bus aiškinama vėliau). Ribojimą įkrauti galima „apeiti“, t.y. įkrovimo metu taikant tinkamą komandą:

- kai sužadinsite komandą įkrauti *Open*, pasirodys pranešimas: *Password Required*, įvesk slaptažodį. Įvesties lauke *Enter database password* klavišais surinkite *expenses*, nuspauskite/laikykite nuspaustą klavišą *Shift* ir bakstelkite *OK*;
- teisingai supraskite kitą informacinį pranešimą: *The database 'Expenses' is read-only, duomenų bazė 'Expenses', tik skaityti. You won't be able to save changes made to data or object definitions in this database*, duomenų bazėje negalėsite įrašyti jokių duomenų ir objektų pakeitimų.

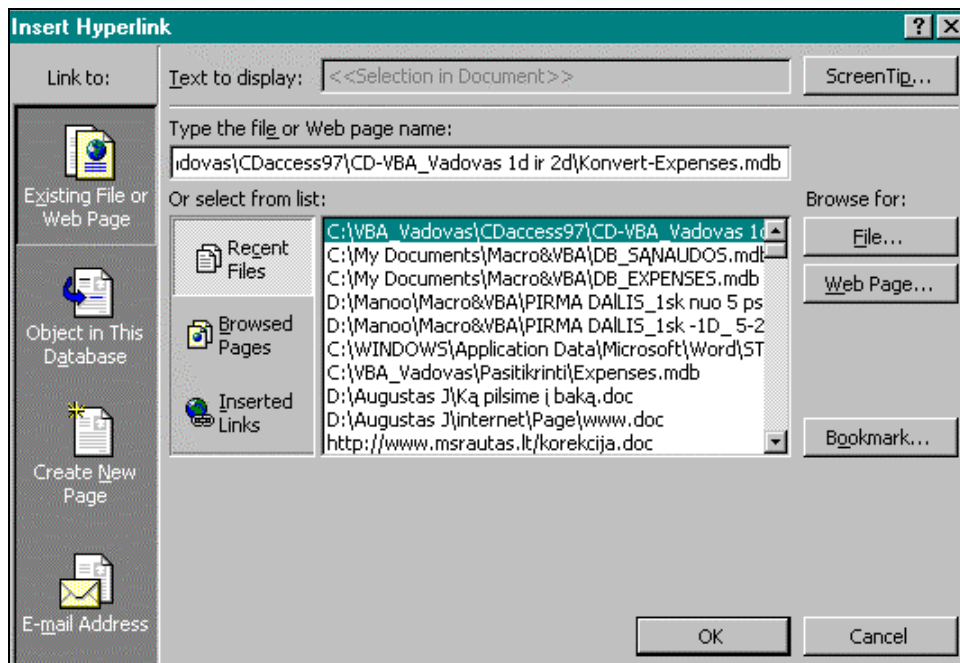
3 PRATIMAS. HIPERSAITO RENGIMAS PANAUDOJANT ATVAIZDĄ *PICTURE*.

Įterptu formoje *frmHyperlinks* valdomuoju elementu - paveikslėliu nustatykite duomenų bazės DB *EXPENSES* hipersaitą su kitos duomenų bazės *Konvert-Expenses* forma *Expense Reports by Employee*:

1. Formos *frmHyperlinks* konstravimo lange nurodykite komandą *Insert → Picture*. Dialogo lange *Insert Picture* nurodykite kelią/adresą, kur yra norimas paveikslėlis, pvz.: rinkmena *balandis dove.vmf*: *C:\Program Files\Microsoft Office\Clipart\Popular\dove.wmf*, rinkmena *lėktuvas Ag00165_.gif*: *C:\Program Files\Microsoft Office\Clipart\Pub60Cor\Ag00165_.gif*. Ir kartu

išbandykime kitą iš anksčiau žinomą vaizdams įkelti komandą, pvz., papūgėlės atvaizdas: Insert → Objects → Microsoft Clip Gallery, skirsnis Animals.

Paveikslėlių matmenis galėsite sumažinti formos lange, kai pažymėsite ir bakstelėsite mygtukų juostoje mygtuką *Properties*, arba paveikslėlį bakstelėsite du kartus, o savybių lango *Image:ImageX* kortelės *Format*, eilutės *Size Mode* įvesties laukelyje išskleisite sąrašą ir nustatysite *Stretch*, ištempti, arba *Zoom*, keisti. Sumažinkite paveikslėlio matmenis, jį nutempkite, ir palikite žemiau *Categories* (1.13 pav.).

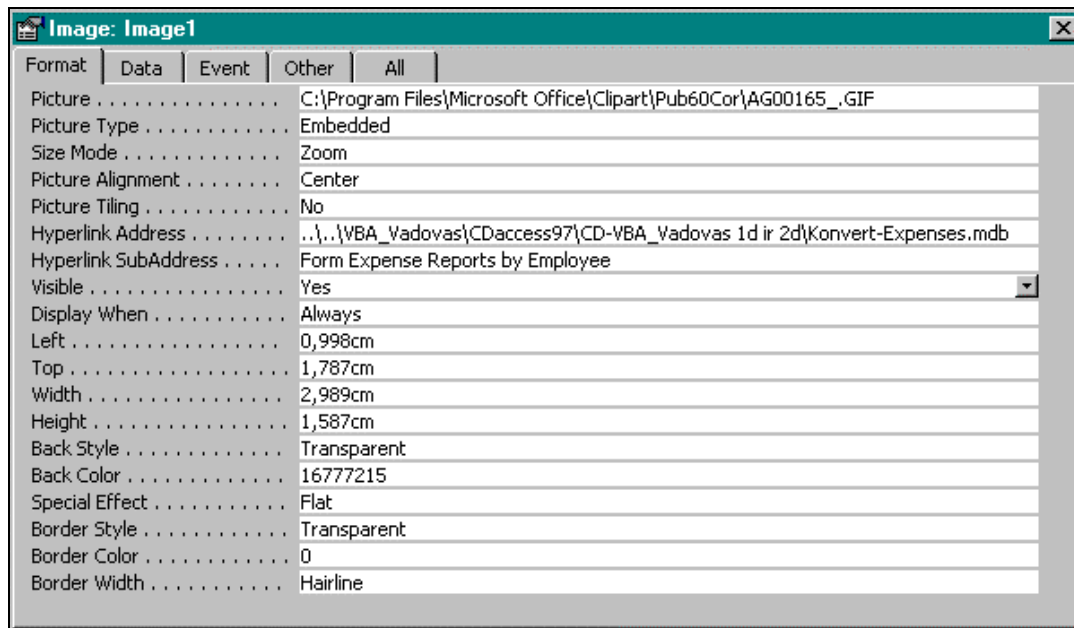


1.11 pav. Hipersaito su duomenų baze *Konvert-Expenses* nustatymo rodinys

Savybių sąrašo lange (1.12 pav.) dabar matysite ne tik visą duomenų bazės rinkmenos *Konvert-Expenses* adresą *Hyperlink Address*, bet taip pat ir papildomą adresą *Hyperlink SubAddress*.

2. Pažymėkite lėktuvo paveikslėlį ir mygtukų juostoje bakstelėkite mygtuką *Properties*. Savybių lange *Image:ImageX* bakstelėkite savybę *Hyperlink SubAddress* ir dešinėje mygtuką su trim taškais. Lango *Insert Hyperlink* stulpelyje *Link to* bakstelkite *Existing File or Web Page*, rinkmenos arba *Web* puslapiai, ir po to skirsnyje *Or select from list* bakstelkite *Recent Files*, paskesnės rinkmenos (1.11 pav.). Pažymėkite kelią į DB rinkmeną *Konvert-Expenses.mdb*. Kadangi reikia saito ne su DB, bet su jos konkrečiu objektu, tad ir reikia jį nurodyti. Lange *Insert Hyperlink* bakstelkit mygtuką *Bookmark*, o naujame lange *Select Place in Document* atverkite *Forms* ir nurodykite tą formą, kurios dabar reikia - *Expense Reports by Employee*.

Pažymėkite papūgėlės atvaizdą, bakstelkit *Properties*, tačiau savybių sąrašo nei *Hyperlink Address*, nei *Hyperlink SubAddress* nerasite. Vadinas, *Clip Art* paveikslėlių formatas hipersaitams nustatyti netinka. Palikime papūgėlę formai pagražinti.



1.12 pav. Formoje *frmHyperlinks* įterpto valdomojo elemento - lėktuvo paveikslėlio savybių rodinys, kai nustatytas hipersaitas su DB *Konvert-Expenses* objektu – forma *Expenses Reports by Employee*

(Dirbant su *Access 97* hipersaitas nustatomas šiek tiek kitaip. Čia pateikiamas hipersaito nustatymo pavyzdys su kitos DB forma. Savybių lange *Image:ImageX* bakstelėkite savybę *Hyperlink Address* ir dešinėje mygtuką su trim taškais. Po to *Insert Hyperlink* lange bakstelėkite mygtuką *Browse*, esantį dešiniau įvesties lauko *Link to file or URL*. Suraskite kitos duomenų bazės rinkmeną *Northwind.mdb*. Dialogo lange *Link to File* nurodykite kelią: *C:\Program Files\Microsoft Office\Office\Samples\Northwind.mdb* ir bakstelėkite *OK*. Lange *Insert Hyperlink* matysite, kokį kelią/adresą surinkote *Link to file or URL* įvesties lauke.

Bakstelėkite mygtuką *Browse*, esantį dešiniau *Named location in file(optional)* įvesties lauko ir dialogo lango *Select Location* kortelėje *Forms* pažymėkite formą *Product List*. Uždarykite langą *Select Location* ir langą *Insert Hyperlink*. Sistema *Access* savybei *Hyperlink Address* priskiria duomenų bazės *Northwind.mdb* kelio/adreso reikšmę, o savybei *Hyperlink SubAddress* duomenų bazės *Northwind* formos *Form Product List* reikšmę.)

3. Įrašykite formą: bakstelėkite mygtuką *Save*. Įkelkite formos vaizdavimo langą ir bakstelėkite valdomąjį elementą - lėktuvo paveikslėlį. Po kiek laiko sistema *Access* vykdydama hipersaitą įkraus duomenų bazę *Konvert-Expenses* ir atvers formą *Expense Reports by Employee*. Kitu atveju, kai bus kitoks hipersaitas, pvz., su DB *Northwind* forma *Product List* ją ir matysite.

4. Uždarykite atvertą formą ir įkrautą DB.

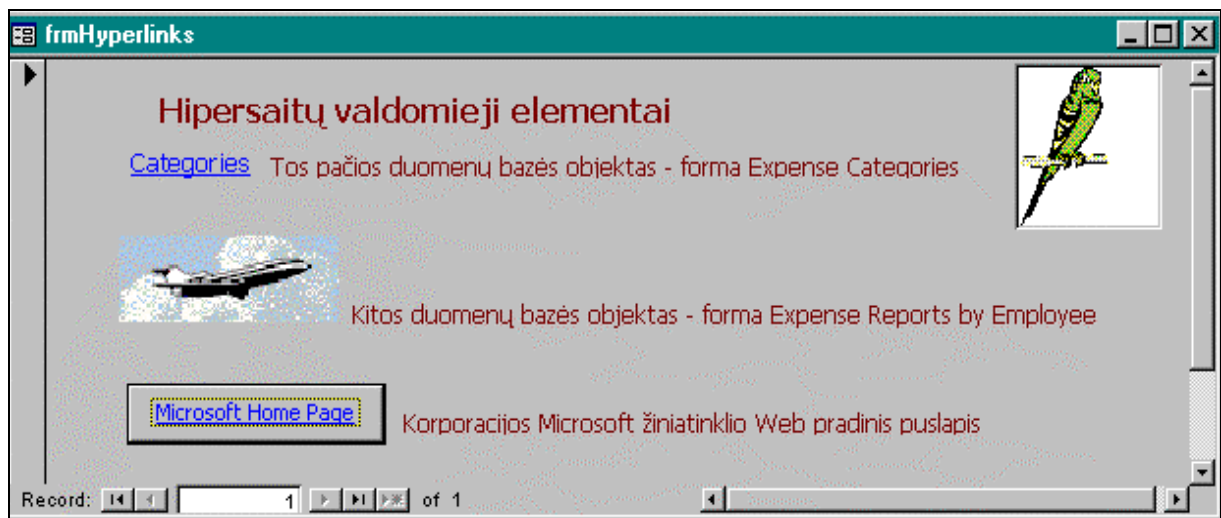
1.07 UŽDUOTIS.

1. Duomenų bazėje DB_ŠAŃAUDOS arba savo automatizuotai suprojektuotoje DB nustatykite pasirinktais kitais paveikslėliais naujai sukurtos formos arba ataskaitos hipersaitą su kitos DB rinkmena, pvz., *Northwind.mdb*, kitoje DB pasirinkta forma ar ataskaita ir su tos pačios DB objektu - forma ar ataskaita.
2. Iš savo darbo aplanko įkraukite atsisiustą DB *Expenses*, (pvz.: *C:\Vadovas_VBA\AtsiustaInfo\expenses.mdb*), atverkite formą *frmHyperlinks*, išbandykite formoje įterptų hipersaitų valdomųjų elementų: *Categories*, paveikslėlio ir mygtuko *Home Page* veikimą.
3. Įkraukite konvertuotą DB *Konvert-Expenses* ir patikrinkite formos *frmHyperlinks* hipersaitų veikimą.

PRATIMAS. HIPERSAITO RENGIMAS PANAUDOJANT KOMANDOS MYGTUKĄ *COMMAND BUTTON*.

Įterptu formoje *frmHyperlinks* valdomuoju elementu – komandos mygtuku nustatykite duomenų bazės DB_EXPENSES hipersaitą su žiniatinklio *Web* puslapiu:

1. Į formos *frmHyperlinks* konstravimo langą įkelkit (jeigu neįkeltas) valdomųjų elementų mygtukų skydelį *Toolbox*. Skydelyje išjunkite, jeigu įjungtas, mygtuką *Control Wizards*, valdomųjų elementų vedliai, ir bakstelėkite mygtuką *Command Button*. Apveskite išivaizduojamo mygtuko kontūrą: nuspauskite kairįjį pelės klavišą formos hipersaitų eilutėje, žemiau paveikslėlio atvaizdo, ir nubrėžkite mygtuko rėmelį. Jeigu po to atsivers dialogo langas *Command Button Wizard*, atsisakykite jo: bakstelėkite *Cancel*. Mygtuko fone matysite tekstą, pvz., *CommandX* (čia X - skaičius, vidinis numeris, kurį įveda sistema).
2. Bakstelkite du kartus mygtuką ir savybių lange *Command Button:CommandX* pakeiskite pavadinimą *Format* kortelės *Caption* eilutėje. Vietoj *CommandX* surinkite *Microsoft Home Page*.
3. Bakstelėkite savybę *Hyperlink Address* ir eilutės dešinėje esantį mygtuką su trim taškais. Įvesties laukelyje *Type the file or Web page name* surinkite *Microsoft* žiniatinklio pradinį adresą *URL*; <http://www.microsoft.com/spustelėkite> *Enter*. Uždarykite savybių langą. Pastebėsite pasikeitusį mygtuko pavadinimą (1.13 pav).
4. Įrašykite formą. Nustatykite formos vaizdavimo langą ir bakstelėkite mygtuką *Microsoft Home Page*. Jeigu kompiuteris prijungtas prie Interneto tinklo, naršyklė išves *Microsoft* svetainės pradinį langą.
5. Formos vaizdavimo lange, kai nustatysite pelės rodyklę į hipersaito ženklą, pasirodys rankos su rodomuoju pirštu rodyklė ir po kiek laiko saito objekto – taikinio vardas arba adresas. Išbandykite naujo hipersaito veikimą.



1.13 pav. Forma *frmHyperlinks* su hipersaitų valdomaisiais elementais

1.08 UŽDUOTIS.

1. Duomenų bazėje DB_SANAUDOS arba savo automatizuotai suprojektuotos DB naujoje formoje įterpkite valdomuosius elementus – du komandų mygtukus nustatyti hipersaitams su norimais skirtingais *Web* puslapiais (vienas iš jų informacijos paieškos, kitas savo nuožiūra).
2. Siekdami palengvinti orientaciją tinkamai apipavidalinkite formą, kurioje išdėstyti visi hipersaitų valdomieji elementai.

3 PRATIMAS.

Parengtą valdomąjį elementą: žymenį, atvaizdą arba mygtuką galima pakeisti kitu. Pažymėkite valdomąjį elementą formos konstravimo lange ir iš komandų sąrašo *Format* nurodykite komandą *Change To*. Papildomajam komandų sąrašui matysite kokių kitų tipų galite pakeisti. Pasidarykite formos *frmHyperlinks* kopiją. DB lange *Forms* ties formos vardu spustelkite pelės dešinįjį klavišą, nurodykite *Copy*, po to *Paste*, pakeiskite vardą, pvz., *Kopija_frmHyperlinks* arba tik *frmHyperlinks1*. Dabar formos kopijoje galima eksperimentuoti - keisti valdomuosius elementus.

1.09 UŽDUOTIS.

1. Įvykdykite 3 pratimą. Keičiant žymenį ar mygtuką paveikslėliu savybės *Picture* reikšmė bus *none*, kadangi atvaizdas neapibrėžtas. Bakstelkit mygtuką *Build* ir nurodykite rastrinį (*bitmap*) paveikslėlį. Keičiant paveikslėlį žymeniu arba mygtuku *Access* naujo tipo valdomojo elemento savybei *Caption* nustato savybės reikšmę *HyperlinkAddress*, jeigu nebuvo nustatyta – savybės reikšmę *HyperlinkSubAddress*. Tikrinkite, įsitikinkite.
2. Suredaguokite formą su atliktais valdomųjų elementų sukeitimais.
3. Patikrinkite hipersaitų veikimą.

Žiniatinklio Web mygtukų juosta. *Access 2000* pildo vidinį hipersaitų panaudojimo protokolą. Judėjimo hipersaitai valdymui paprastinti numatyta mygtukų juosta *Web*. Mygtukas *Back*, atgal, grąžina ankstesnį hipersaitą, *Forward*, pirmyn, - į kitą iš eilės (bet tik į jau panaudotą hipersaitą). Kai norima atsisakyti jau sužadinto hipersaito, reikia bakstelėti *Stop*.

Exp Rpt Name	Date Subn	Advance	Total Expenses	Amount Due
Feb. '95 Sales Trip	1995.03.01	0,00 Lt	464,00 Lt	464,00 Lt

1.14 pav. Atlikus įrašų reikšmių taisymus lentelėje *Employees* formos *Expense Reports by Employee* įrašai pasikeičia/atsinaujina automatiškai. Pateikiamas formos pagrindinės dalies atsinaujinusio pirmo įrašo rodinys (žr. žemiau 1.10 UŽDUOTIS)

1.10 UŽDUOTIS.

1. Patikslinti ir sutvarkyti anksčiau skubotai užpildytus įrašus. Tarkime, išryškėjo, kad apsirikote (taip ir buvo), įrašydami reikšmes supainiojote laukus (jeigu taip nėra - apsirikite sąmoningai dabar ir ištaisykite apsirikimą), reikia ištaisyti apsirikimus. Kartu

prisiminsite arba sužinosite, kaip tai daroma, nes tokie apsirikimai galimi. Pirmas patarimas - neskubėti visko naikinti ir kurti iš naujo, bet pagalvoti, kaip taisyti.

2. Palyginkite matytus anksčiau formos *Expense Reports by Employee* rodinius ir dabar pateikiamą 1.14 pav. Pagrindinėje formos dalyje pamatykite skirtumus. Tenka pripažinti, kad ankstesnėje formoje buvo akivaizdžių netikslumų. Panašiai atsitinka kopijuojant iš kitos DB lentelės. Tarkime, apsirikimai išryškėjo tik dabar, kai norima išvesti prasmingas kai kurias reikšmes, todėl teko atidžiau peržvelgti įrašus.
3. Lentelės *Employees* įrašų lange kopijuokite vieno lauko (stulpelio) visas reikšmes iš karto ir įterpkite kitame. Jeigu reikia, įterpkite laikinus (tarpinius) laukų stulpelius, šalinkite nereikalingus. Kartais gali tekti keisti laukų savybes lentelės *Field Properties* srityje. Duomenų tipų keitimo galimybės dažniausiai būna ribotos.
4. Kadangi lentelė yra formos pagrindinės dalies įrašų šaltinis, tai formoje valdomųjų elementų reikšmės pasikeičia automatiškai. Formos iš naujo sukurti nereikia.
5. Išbandykite savo sugebėjimus. Nors čia dar joks programavimas, tačiau viena kita panaši ir neprogramavimo užduotis yra reikalinga net tam, kad įsitikintum - moki, žinai, gali ar ne atlikti ir tokius pasirengimo darbus.

1.6. Paslaugų programa **Combo Box Wizard**

Kai formoje daug įrašų, duomenų paieškai reikalinga papildoma priemonė. Paslaugų programa *Combo Box Wizard* gali parengti kombinuotąjį bloką *Lookup*, keitiniai, kuris gali padėti išrinkti norimą reikšmę iš sąrašo, o sistema *Access* automatiškai gali išvesti reikiamą įrašą.

PRATIMAS.

Panaudojant paslaugų taikomąją programą *Combo Box Wizard* reikia parengti kombinuotąjį sąrašą/bloką, vaizduojantį darbuotojų pavardes abėcėlės tvarka ir procedūrą, kuri vykdoma išrenkant pavardes iš sąrašo, randa ir išveda į ekraną konkrečią pavardę.

1 Duomenų bazės lange *DB_EXPENSES:Database* atverkite formą *Expense Reports by Employee* konstravimo lange. Įkelkite mygtukų skydelį *Toolbox*. Patikrinkite ar įjungtas mygtukas *Control Wizards*. Suraskite mygtuką *Combo Box* ir formos darbo srityje bakstelėkite eilutėje, dešiniau mygtuko *Close Form*.

2. Ekrane pasirodys pirmasis langas *Combo Box Wizard*, kuriame pranešama: *This wizard creates a combo box, which displays a list of values you can choose from. How do you want your combo box to get its values? (Meistras parengia kombinuotąjį sąrašą, iš kurio galėsite pasirinkti reikšmes. Kaip pateikti reikšmes?)* Žemiau yra trys pasirenkamieji jungikliai:

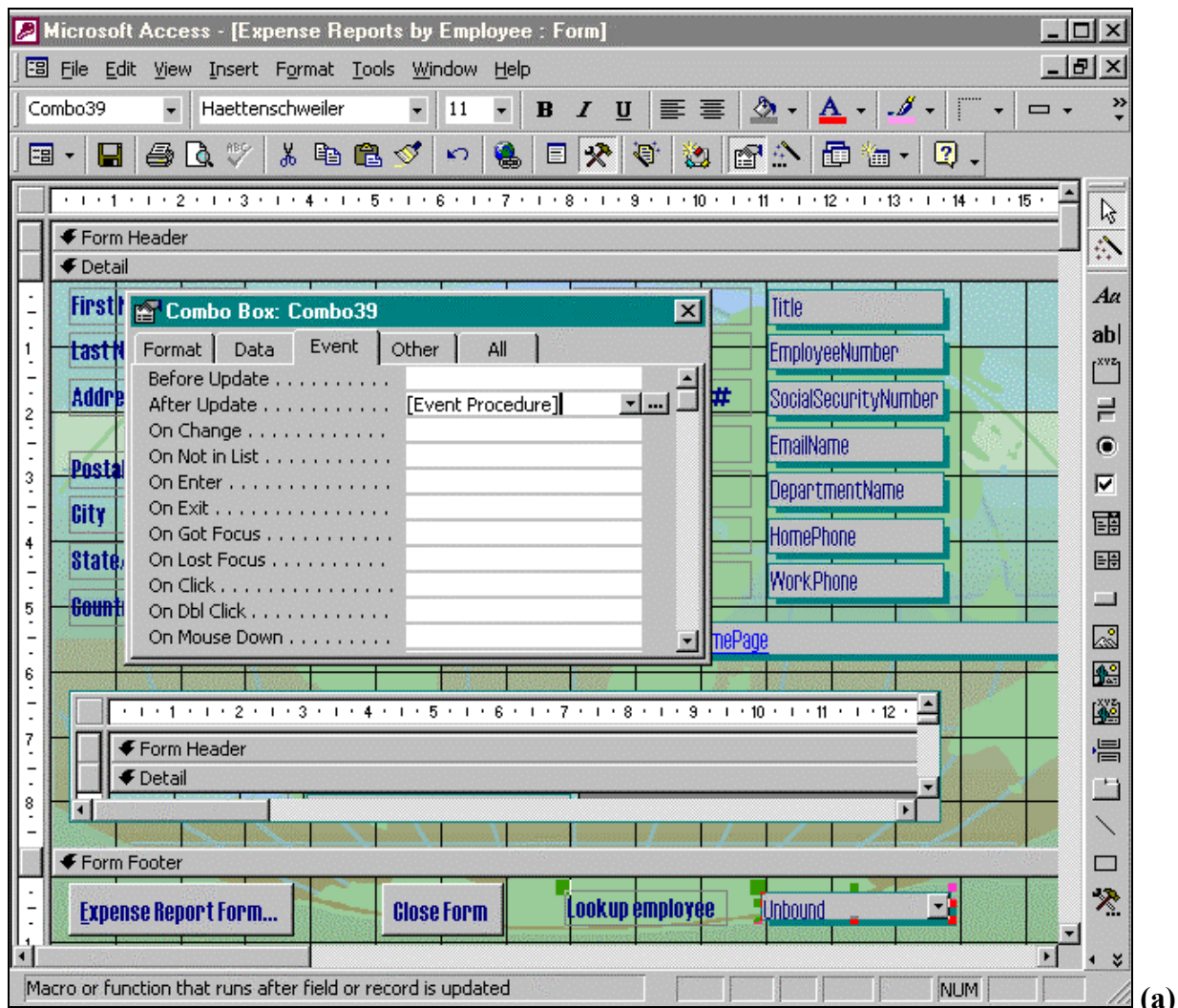
- I want the combo box to look up the values in a table or query, noriu, kad reikšmės būtų pateiktos iš lentelės arba užklauso.
- I will type in the values that I want, įrašysiu reikšmes, kokias norėsiu.
- *Find a record on my form based on the value I selected in my combo box*, įrašo paieška formoje pagal reikšmę, pasirenkama kombinuotajame sąrašo.

Įjunkite trečiąjį jungiklį ir bakstelėkite mygtuką *Next*.

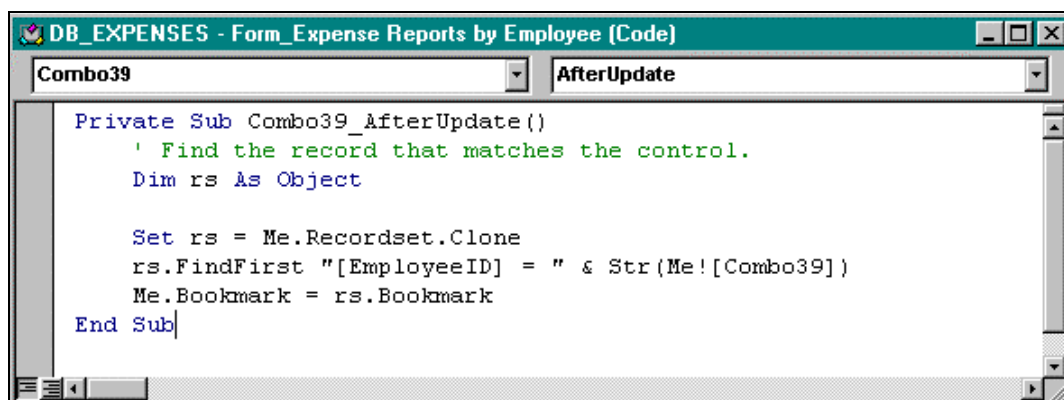
3. Antrajame lange *Combo Box Wizard* klausiama: *Which fields contain the values you want included in your combo box?* (Kuriuose laukuose yra reikalingos reikšmės?) The fields you select become columns in your combo box. (Laukai, kuriuos pažymėsite, taps sąrašo stulpeliais).

Nors sąrašo bus rodomos tik laukų *LastName* ir *FirstName* reikšmės, laukas *EmployeeID* taip pat reikalingas, kadangi yra pirminio rakto laukas. Perkelkite *EmployeeID*, *LastName* ir *FirstName* iš sąrašo *Available Fields*, esami laukai, į sąrašą *Selected Fields*, išrinkti laukai.

Pažymėkite po vieną lauką ir bakstelkite mygtuką, pažymėtą dešinine varnele ">". Bakstelėkite *Next*. Paslaugų programa *Combo Box Wizard* parengia *VBA* procedūrą. Reikšmė *EmployeeID* panaudojama įrašui išvesti pagal nurodomą pavardės reikšmę.



(a)



(b)

1.15 pav. Paslaugų programa *Combo Box Wizard* parengė kombinuotąjį sąrašą *Lookup Employee* su reikšmių lauku *Unbound* (a) ir įvykio *AfterUpdate* apdorojimo procedūrą *Combo39_AfterUpdate* (b)

4. Trečiajame lange *Combo Box Wizard* galima pakeisti sąrašo išorinį vaizdą. Klausima: How wide you like the columns in your combo box? (Kokio pločio turi būti stulpeliai ?) To adjust the width of a column, drag its right edge to the width you want, or double – click the right edge of the column heading to get the best fit (stulpelio plotį pakeisite vilkdami stulpelio pavadinimo juostoje

dešiniąją vertikaliąją liniją, arba ją bakstelėkite du kartus – tada nusistatys minimalus stulpelio plotis).

Ijunkite jungiklį, jeigu jis neįjungtas - *Hide key column*, paslėpti raktinį stulpelį. O jeigu įjungtas – išjunkite, ir vėl įjunkite – pamatysite jo veikimą. Bakstelėkite *Next*.

5. Ketvirtajame lange *Combo Box Wizard* klausia: *What label would you like for your combo box?* (Kokiu vardu pavadinti sąrašą?) Surinkite *Lookup employee*, darbuotojo paieška. Žemiau sakoma: *Those are all the answers the wizard needs to create your combo box*, tai viskas, ką norėjo patikslinti sąrašo rengimo vedlys. Bakstelėkite mygtuką *Finish*, Formos *Expense Reports by Employee* lange pamatysite du valdomuosius elementus: kombinuotojo sąrašo (*combo box*) pavadinimą *Lookup employee* ir greta sąrašo reikšmių lauką, kuriame matomas tekstas – *Unbound*, nesusietas. Dabar čia sąrašo dar nematysite, nes čia formos konstravimo langas (1.15 pav., (a)).

6. Įrašykite formą, nustatykite formos vaizdavimo langą *Form View*, išskleiskite sąrašą, matysite trijų darbuotojų pavardžių ir vardų sąrašą. Kai sąrašą bakstelėsite pasirinktą pavardę, formoje iš karto bus išvedamas to darbuotojo įrašas. Grįžkite vėl į formos konstravimo langą *Design View*. Peržiūrėkite valdomųjų elementų savybių langus.

7. Bakstelėkite du kartus ties *Lookup employee*. Ekrane pamatysite savybių langą *Label: Lookup employee_Label*. Peržiūrėkite korteles. Kortelėje *All* matysite, kad vardo savybės *Name* reikšmė – *Lookup employee_Label*, o pavadinimo *Caption* reikšmė – *Lookup employee*. Uždarykite tyrinėtą savybių langą. Bakstelėkite du kartus sąrašo reikšmių lauką *Unbound*, ekrane matysite savybių langą *Combo Box:ComboX*. Atkreipkite dėmesį į savybės *Row Source*, eilutės šaltinis. Tai *SQL* kalba parašyta (rašoma būtinai vienoje eilutėje) šaltinio - lentelės *Employees* dalies išraiška:

```
SELECT [Employees].[EmployeeID], [Employees].[FirstName], [Employees].[LastName]  
FROM Employees;
```

Paskaitykite ir iššifruokite išraišką.

Kaip pamatyti procedūrą, kurią parengė paslaugų programą? Atverkite kortelę *Event*, įvykis, ir savybės eilutės *After Update* dešinėje bakstelėkite mygtuką su trimis taškais. Jeigu procedūros lange nustatyta veikseną *Procedure View* (mygtukas lango apačioje kairėje), matysite procedūros turinį (1.15 pav.(b)).

Įvykiai *AfterUpdate*, po atnaujinimo, ir *Before Update*, prieš atnaujinimą, yra sudėtingesni, negu anksčiau nagrinėti – *Click*. Kombinuotasis blokas *Lookup employee* atpažįsta įvykį *After Update* tuoj pat, kai sąrašą nurodoma tarnautojo pavardė. Procedūra sinchronizuoja formą su nurodoma sąrašą pavardės ir vardo reikšme ir išveda darbuotojo įrašą.

1.7. Paslaugų programa **Switchboard Manager**

Paslaugų programa *Switchboard Manager* skirta programai *Database Wizard* sukurtai valdymo formai *Switchboard* modifikuoti. Be to, jeigu programa *Database Wizard* ir nebuvo naudojama, paslaugų programa *Switchboard Manager* gali sukurti naują formą *Switchboard*.

Main *Switchboard* lange bakstelėkite *Expense* komutacinio skydelio mygtuką *Change Switchboard Item*.

Switchboard Manager lange matysite trijų puslapių sąrašą *Switchboard Pages: Main Switchboard (Default), Forms Switchboard* ir *Reports Switchboard*. Lentelės ir užklauskos vartotojo sąsajose nedalyvauja (jose valdymo mygtukai neįtaisomi). Vaizduojamo lango dešinėje matomi konstravimo ir redagavimo mygtukai: *Close, New, Edit, Delete, Make Default*.

- Mygtukas *New* skirtas naujam komutatoriui – naujai valdymo skydelio formai kurti.
- *Edit* – esančiam komutatoriui modifikuoti. Atvertame dialogo lange *Edit Switchboard Page* vardijami esamo komutatoriaus mygtukų pavadinimai. Čia komutatorių galima

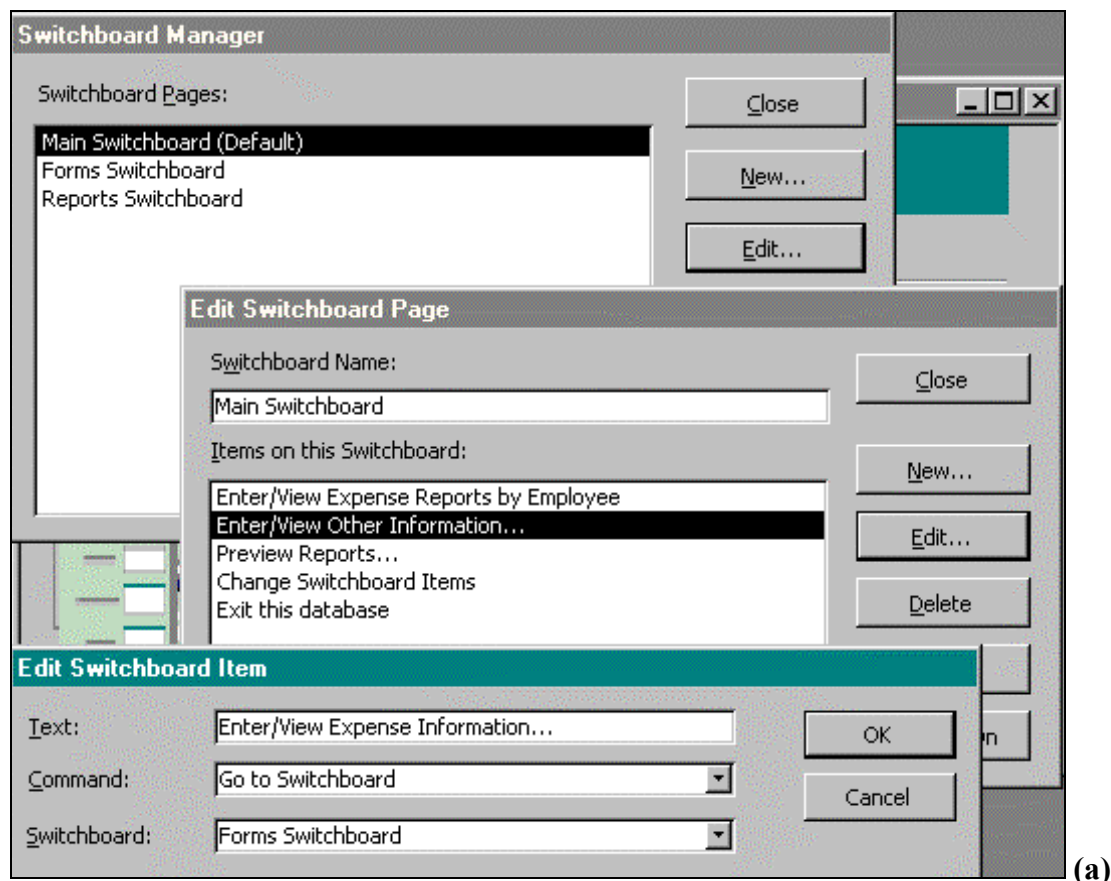
papildyti nauju mygtuku, šalinti netinkama, keisti mygtuko pavadinimą ir keisti išdėstymą.

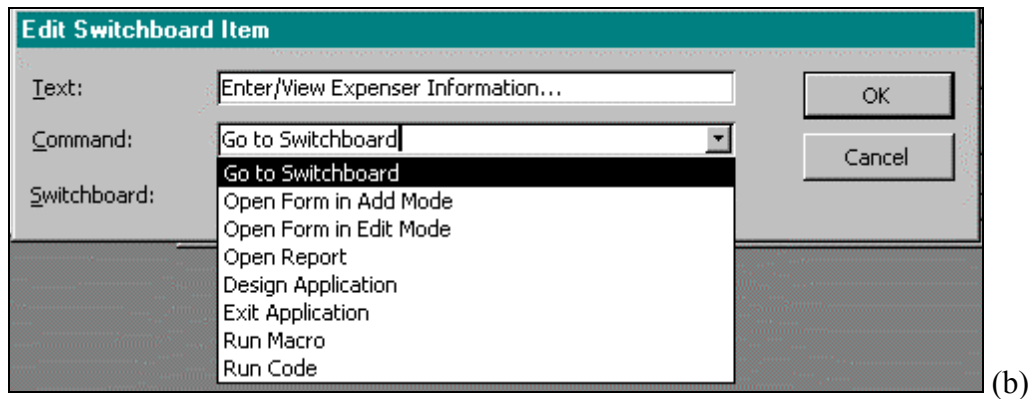
- *Delete* – šalina pažymėtą sąrašą *Switchboard Pages* puslapį.
- *Make Default* – keičia pažymėto sąrašą puslapio parametrus standartiniais.

PRATIMAS.

Pagrindinio komutacinio skydelio *Main Switchboard* antrojo mygtuko pavadinimą *Enter/View Information* pakeiskite prasmingesniu pavadinimu – *Enter/View Expense Information*, sąnaudų įvestis ir peržiūra.

1. Bakstelėkite mygtuką *Change Switchboard Items*. Dialogo lange *Switchboard Manager* pažymėkite *Main Switchboard* ir bakstelėkite mygtuką *Edit*. Dialogo lango *Edit Switchboard Page* sąrašą *Items on this Switchboard* pažymėkite *Enter/View Other Information* ir bakstelėkite *Edit*. Dialogo lango *Edit Switchboard Item* įvesties lauke *Text* redaguokite pavadinimą. Vietoj *Other* surinkite *Expense*. Peržiūrėkite sąrašus *Command* ir *Switchboard*, bet nieko nekeiskite. Bakstelėkite *OK*. Bakstelėkite *Close* ir dar kartą *Close*. Dabar *Main Switchboard* lange matysite antrojo mygtuko pakeistą pavadinimą.





(b)

1.16 pav. Esamo komutatoriaus mygtuko pavadinimo *Enter/View Other Information* keitimas į *Enter/View Expense Information*

Lange *Edit Switchboard Item* sąrašas *Command* leidžia nustatyti (keisti) komutatoriaus mygtuko veiksmą pasirenkant vieną iš galimų 8. Duotuoju atveju nusistatęs savaime *Go to Switchboard* turi atveri komutatoriaus kitą vidinę mygtukų formą *Forms Switchboard*, kuri nustatoma šiuo atveju taip pat savaime trečiajame sąrašė *Switchboard*.

2. Dar kartą pakartokite tuos pačius veiksmus, tik šį kartą surinkite antrojo mygtuko pavadinimą lietuviškai - Sąnaudų įvestis ir peržiūra. Išbandykite mygtuko veikimą, po to vėl grąžinkite buvusį anglišką – *Enter/View Expense Information*.

Lange *Switchboard Manager* pasirinkus mygtuką *New* galima kurti naują komutatorių.

3. Baikite darbą.

1.11 UŽDUOTIS.

1. Pakeisti automatizuotai suprojektuotos duomenų bazės formų ir ataskaitų pavadinimus (*Caption*). Pakeisti komutatoriaus skydelio mygtukų angliškus pavadinimus lietuviškais pagal formas ir ataskaitas.
2. Patikslinti ir naujai suredaguoti duomenų bazės DB_SĄNAUDOS valdymo komutatoriaus formą.
3. Išbandyti komutatoriaus veikimą.

1.8. Vartotojo sąsajų valdymas

Sistema *Access* turi numatytas priemones ir gali keisti vartotojo kontaktavimo būdus su duomenų baze ir paslaugų programomis. Pavyzdžiui, jeigu su DB dirbs nepatyręs vartotojas, reikia iš anksto pasirūpinti duomenų apsauga, taikyti nustatymus, kad nepasirodytų komandos, kuriomis galima ką nors pakeisti. Sistema *Access* turi specialias priemones vartotojo sąsajoms valdyti ir duomenų apsaugai.

Apžvelkime trijų rūšių priemones, taikomos be programavimo duomenų bazei apsaugoti, valdyti ir automatizuoti:

- Sistemos *Access* veikimą galima keisti nustatant iš anksto paleidimo charakteristikas: neleisti arba leisti išvesti duomenų bazės langą, kai pradeda veikti taikomoji programa, atšaukti klavišų kombinacijas duomenų bazės langui išvesti *F11* arba *Alt+F1*. Tačiau atšaukti klavišo *Shift* veikimą, kuris sudaro galimybę ignoruoti paleidimo charakteristikas, be programavimo negalima.
- Nesankcionuotas kreiptis į duomenų bazę galima uždrausti slaptažodžiu. Duomenų bazę galės atverti tik žinantis slaptažodį, tačiau tai nesaugo duomenų, jeigu vartotojas elgsis nekorektiškai, pvz., dėl nepakankamo išprusimo.

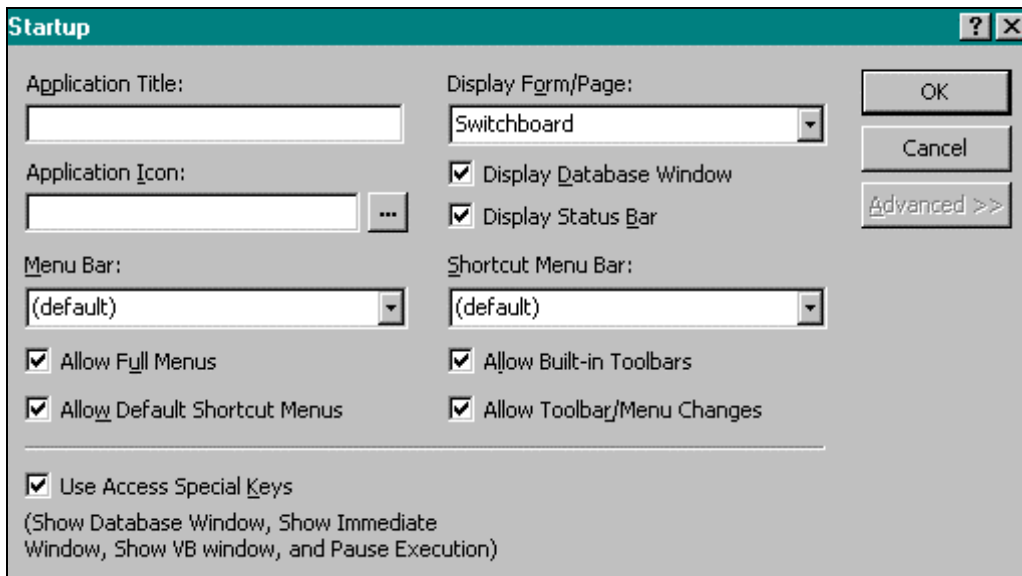
- Į komandų sąrašus ir mygtukų juostas galima įterpti savas komandas ir mygtukus arba kurti savus komandų ir mygtukų darinius, taikomosioms programoms valdyti. Tačiau neprogramuojant galima pasinaudoti tik sistemos standartinėmis įterptinėmis komandomis. Vartotojo komandų procedūras, valdomas klaviatūra galima parengti makroprogramavimo būdu.

1.9. Paleidimo parametrų nustatymas

Duomenų bazės lange iš komandų sąrašo *Tools*, priemonės, nurodykite komandą *Startup*, pradėti veikti. Dialogo lange *Startup* bakstelėkite mygtuką *Advanced*, žvalgomasis. Dabar *Startup* lange matysite visas galimas parinktis (1.17 pav.). Programos/prietaikos pavadinimo *Application Title* ir ženklo *Application Icon* pakeitimą matysite tuoj pat, kai tik uždarysite langą *Startup*, kitų parametrų veikimą – kai pakartotinai paleisite *Access*.

a. Paleisties (*Startup*) parametrų lentelė (žr.: 1.17 pav.)

<i>Application Title</i>	Programos/prietaikos pavadinimo tekstas. Jeigu nenustatytas, išvedamas standartinis - Microsoft Access
<i>Application Icon</i>	Programos/prietaikos ženklas. Jeigu nenustatytas, išvedamas standartinis - Microsoft Access
<i>Display Form/Page</i>	Po paleisties ekrane išvedamos formos ar puslapio vardas. Išvedama komutatoriaus forma, jeigu jį sukurta panaudojant Database Wizard
<i>Display Database Window</i>	Rodo, ar reikia išvesti DB pagrindinį langą
<i>Display Status Bar</i>	Ar išvesti būsenos eilutę
<i>Menu Bar</i>	Rodo, kurį meniu išvesti vietoj standartinio visuose prietaikos languose išskyrus tas formas arba ataskaitas, kurioms specialiai tie meniu parengti
<i>Shortcut Menu Bar</i>	Rodo, kurį parankinį meniu išvesti vietoj standartinio visuose prietaikos languose išskyrus tas formas arba ataskaitas ir formų valdomuosius elementus, kurioms(-ems) specialiai tie meniu parengti
<i>Allow Full Menus</i>	Rodo, kad Access turi išvesti visą meniu ar trumpinamą komandų ir punktų meniu, kuriame nėra prietaikos keitimo komandų
<i>Allow Default Shortcut Menus</i>	Rodo ar Access turi išvesti visą standartinį parankinį/pagal kontekstą meniu
<i>Allow Built-in Toolbars</i>	Ar Access turi išvesti visas standartines priemonių (mygtukų) juostas. Vartotojo parengtoms išvesti, reikia mygtuką išjungti
<i>Allow Toolbar/Menu Changes</i>	Ar gali vartotojas modifikuoti priemonių juostas. Kai išjungtas, blokuojamas pelės dešinys klavišas, juostų mygtukas Close ir komanda Toolbars. Juostų matmenis, padėti, matymą keisti galima
<i>Use Access Special Keys</i>	Ar turi Access reaguoti spustelint standartinius greitojo reagavimo klavišus: F11 arba Alt + F1 (atverti, atstatyti langą <i>Database</i>); Ctrl + G (atverti langą <i>Immediate</i>); Alt + F11 (redaktorius <i>VB</i>); Ctrl + F11 (pereiti iš standartinės į vartotojo menu eilutę); Ctrl + Break (nutraukia programos - <i>VBA</i> procedūros vykdymą, nutraukia projekto vykdymą ir įrašų skaitymą iš serverio)



1.17 pav. Dialogo lange *Startup* galima nustatyti DB paleisties parametrus

PRATIMAS.

Patyrinėkite paleidimo parametrų nustatymą ir panaudojimą. Pakeiskite *Access* taikomosios programos pavadinimą ir pasidomėkite apsauga nuo nesankcionuotos kreipties.

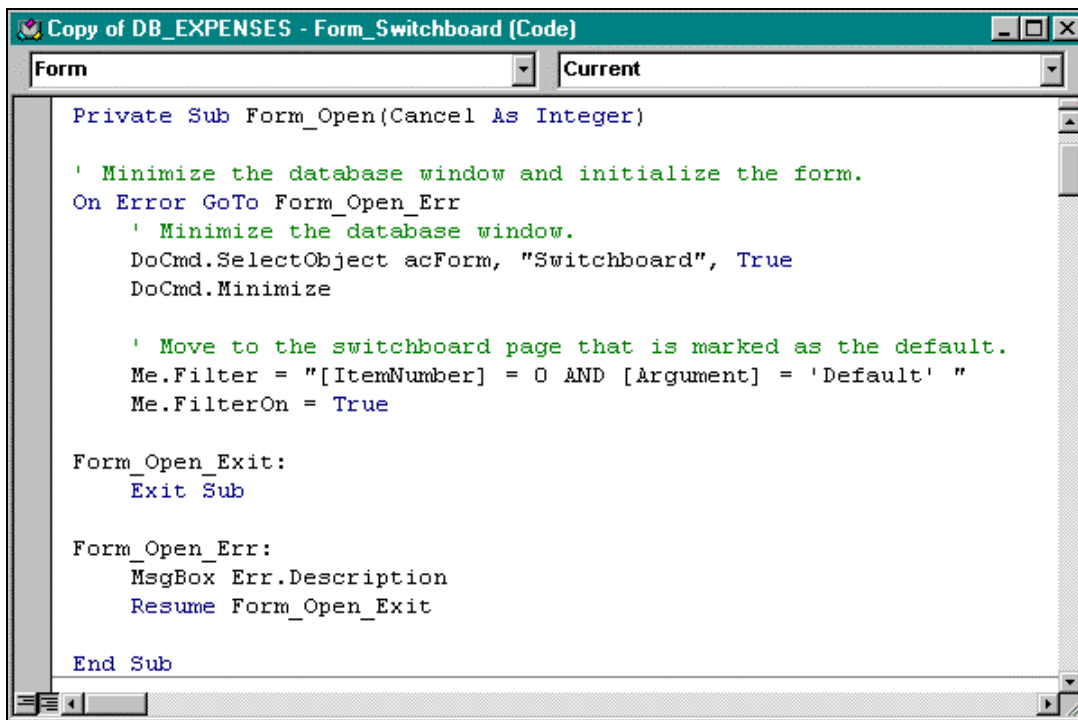
1. Komutacinio skydelio *Main Switchboard* arba duomenų bazės lange nurodykite: *Tools* → *Startup* → *Advanced*
2. *Startup* lange įvesties laukelyje *Application Title*, taikomosios programos pavadinimas, surinkite *Expenses*.
3. *Startup* lange bakstelkite mygtuką *Advanced*, išjunkite visus nustatomuosius jungiklius ir palikite įjungtą tik vieną *Display Status Bar*, vaizduoti būsenos eilutę.
4. Bakstelėkite *OK*. Pastebėkite, kad programos pavadinimas pakito iš karto į *Expenses*. Jeigu pradžioje buvote pagrindinio komutacinio skydelio lange ir programos pavadinimas buvo *Microsoft Access* (žr. programos lango ir su juo sutapdinto dokumento lango pavadinimas *Microsoft Access* – [*Main Switchboard*]), tai dabar vietoj *Microsoft Access* matote – *Expenses*. Panašų pasikeitimą matysite duomenų bazės lange ir kt.

Jeigu programos langas nesutapdintas su dokumento langu, tai programos lango pavadinimas dabar ne *Microsoft Access*, bet tik *Expenses*. Pamatykite sutapdintus ir nesutapdintus langus ir jų pavadinimus.

Startup lange galite grąžinti ankstesnius parametrus, nustatytuosius pagal nutylėjimą ir dar kartą peržiūrėkite langus.

Rengiant duomenų bazę *Expenses*, paslaugų programa *Database Wizard* parengia procedūrą, kuri vykdoma pirmą kartą atveriant formą *Switchboard*. Joje yra operatoriai, išvedantys į ekraną ir suglaudžiantys duomenų bazės langą. Procedūra vykdoma po to kai paleidžiama veikti programa *Access*, bet būtinai atsižvelgiant į *Startup* parametrus. Kadangi dabar veikia ankstesnis *Access* sistemos paleidimas (*Startup* parametrus pakeitėme tik dabar) norėdami pamatyti nustatytų parametrų įtaką, modifikuokime procedūrą taip, kad nebūtų vaizduojamas duomenų bazės langas.

5. Nustatykite formos *Main Switchboard* konstravimo veikseną *Design View*, matysite formą *Switchboard*, bakstelėkite mygtuką juostoje mygtuką *Properties*. Formos savybių lange *Form* bakstelėkite savybę *On Open* ir toje eilutėje, dešinėje mygtuką su trim taškeliais. Ekrane pamatysite langą - *Form Switchboard: Class Module* procedūros, apdorojančios įvykį *On Open*, kai atveriamą formą. Lango apačioje, kairėje, turi būti įjungtas mygtukas *Procedure View*.



1.18 pav. Duomenų bazės lango išvedimo ir sumažinimo operatoriai įvykio *Open* procedūroje

Trys procedūros eilutės (neskaitant žalia spalva išvestų komentarų eilučių, žemiau eilutės *Minimize the database window* yra ir komandų eilutės, išvedančios ir minimizuojančios duomenų bazės langą. Pašalinkite tas tris procedūros ir dvi komentarų eilutes. Pažymėkite eilutes pradedant komentaro eilute *Minimize the database window* ir baigiant procedūros eilute *DoCmd.Minimize* ir spustelėkite klavišą *Delete*.

6. Įrašykite pakeitimą: bakstelėkite mygtuką *Save* ir uždarykite formą *Switchboard*. Uždarykite duomenų bazės langą – *Close*.

7. Vėl atverkite duomenų bazės langą iš *File*, nurodykite vardą *Expenses*. Dabar taikomoji programa *Expenses* bus paleista be duomenų bazės lango piktogramos, be mygtukų juostos, turės supaprastintus komandų sąrašus. Negalėsite duomenų bazės lango iškviesti klavišais *F11* arba *Alt + F1*. Jeigu dabar dar kartą uždarysite *Main Switchboard* langą, *Expenses* programos lange iš *File* galėsite nurodyti, tik *Exit*, baigti darbą.

8. Paleiskite vėl *Access*. Iš *File* pažymėkite programą *Expenses*, nuspauskite (nuspauskite ir laikykite) klavišą *Shift* ir bakstelėkite mygtuką *OK*. Matysite, kad anksčiau nustatyti parametrai *Startup* nebegalioja. Vėl susigrąžinsite pirminį *Main Switchboard* ir duomenų bazės vaizdą, anksčiau nustatyti parametrai *Startup* lange liko apeiti ir neburi įtakos. Kaip laikinai atsisakyti nuspausto klavišo *Shift* veikimo, sužinosite vėliau (3 dalis, 19 skyrius).

9. Paleiskite duomenų bazę dar kartą apeidami nustatytus *Startup* parametrus. Išveskite dialogo langą *Startup*, vėl įjunkite visus jungiklius. Uždarykite ir vėl įkraukite duomenų bazę. Vėl matysite ankstesnę duomenų bazę – tokią, kaip prieš pratimo pradžią.

1.12 UŽDUOTIS.

1. Įkrauti atsisiųsta DB *Expenses.mdb* (slaptažodis *expenses*) vieną kartą kaip įprasta, kitą kartą „apeinant“ paleisties parametrus. Išryškinti ir palyginti DB valdymą vienu ir kitu atveju.
2. Įkrauti konvertuotą DB *Konvert-Expenses*, įsitikinti ar veikia formoje *frmHyperlinks* hipersaito valdomasis elementas balandžio atvaizdas. Konvertuojant kartais netiksliai nurodomas adresas. Kai hipersaitas veikia, kodėl po įkrovimo pirmiausia ekrane matomas formos *Product List*, o ne kokio nors kito objekto langas? Sužinoti įkraunamos

DB adresą, jeigu reikia, suredaguoti. Paaiškinti, kaip tą padaryti. Kodėl pirmiausia ekrane pasirodo atvertas formos langas?

3. Kai veikiant minėtam hipersaitui bus įkrauta DB, ekrane išvesti komutatoriaus formą ir susipažinti su DB pagrindiniais objektais panaudojant komandų mygtukus.

1.10. Apsauga nuo nesankcionuotos kreipties

Pirmas būdas. Kai dirbate su duomenų baze (programa), kiti vartotojai negalės kreiptis į duomenų bazę, kai prieš įkraudami nustatysite režimą *Exclusive*, išskirtinė kreiptis. Verti dėmesio ir kiti pasirinkimai (1.19 pav.).

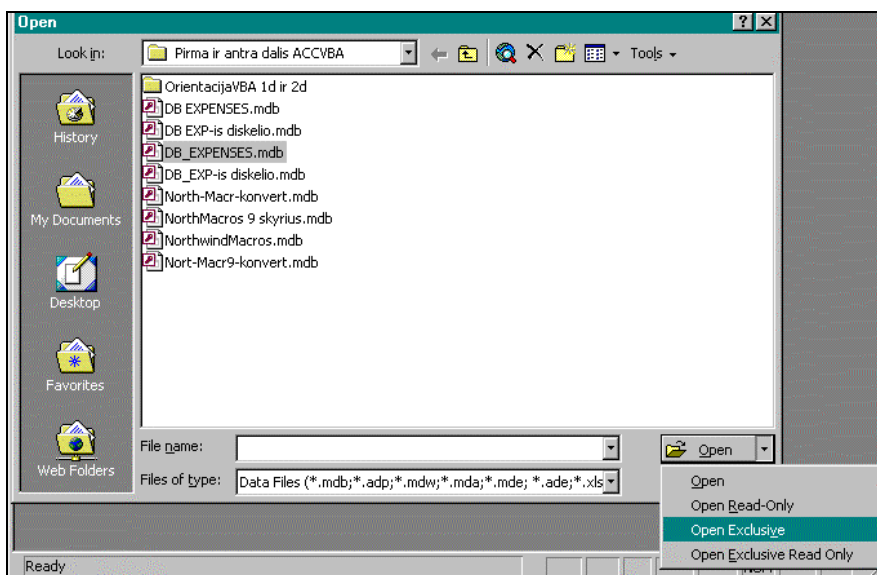
PRATIMAS.

Paleiskite veikti sistemą *Access* ir duomenų bazę įkraukite iš komandų sąrašo *File* nurodydami komandą *Open* (1.19 pav.). Dialogo lange pažymėkite *DB_EXPENSES.mdb*, įjunkite jungiklį *Exclusive*, išskirtinė kreiptis, nuspauskite *Shift* ir bakstelėkite mygtuką *Open*. Nuspaustas klavišas *Shift* leidžia nepaisyti nustatytų parametrų lange *Startup*. Dabar darbo metu, t.y. kol atverta duomenų bazė *DB_EXPENSES*, kiti vartotojai negalės kreiptis į duomenų bazę. Tai paprasčiausia apsauga, bet veikia tik tada, kai dirbate su duomenų baze.

Antras būdas. Kiti vartotojai negalės kreiptis į duomenų bazę, kai ją apsaugosite panaudodami slaptažodį. DB atverkite taip, kaip ką tik patarta aukščiau (Pirmas būdas).

PRATIMAS.

Duomenų bazės lange įveskite slaptažodį: *Tools* → *Security*, apsauga, → *Set Database Password*, slaptažodžio nustatymas. Dialogo lange *Set Database Password* laukelyje *Password* surinkite slaptažodį, pvz., *expenses*. Šį kartą žodį *expenses* surinkite mažosiomis raidėmis (žinokime, kad slaptažodyje mažųjų ir didžiųjų raidžių reikšmės skiriasi). Laukelyje *Verify*, tikrinti, pakartokite slaptažodį ir bakstelėkite *OK*. Uždarykite duomenų bazės langą *DB_EXPENSES: Database* ir vėl atverkite. Kai norėsite atverti, kaip įprasta, ekrane pasirodys slaptažodžio įvesties langas *Password Required*, įveskite dabar žinomą slaptažodį *expenses* ir *OK*. Šį kartą, kaip supratote veikia *Startup* nustatytieji parametrai ir duomenų bazės lango ekrane nematote. Atsisakykite: *Exit this database* ir vėl paleiskite, ignoruodami anksčiau nustatytus *Startup*, bet dabar nepageidautinus parametrus: *File* → nuspauskite *Shift* → *DB_EXPENSES*. Įveskite slaptažodį *expenses*, vėl nuspauskite *Shift* ir *OK*. Vėl matysite visą duomenų bazės lango vaizdą. Valdymo skydelį įkelsite: *Forms* → *Switchboard* → *Open*.



1.19 pav. Galimi duomenų bazės įkrovimo atvejai taikant komandą *File* → *Open*

Apsauga panaudojant slaptažodį nėra sudėtinga, tik reikia nepamiršti slaptažodžio, tačiau tai nevisiška apsauga.

PRATIMAS. KAIP ATISISAKYTI SLAPTAŽODŽIO?

Duomenų bazės lange: *Tools* → *Security* → *Unset Database Password*, surinkite slaptažodį *expenses* ir OK. Gali pasirodyti pranešimas: *You need to have the database open for exclusive use to set or unset the database password* (slaptažodį nustatyti arba jo atsisakyti galėsite tik išskirtinės kreipties atveju). To open the database exclusively, close the database, and then reopen it using the *Open Database* command on the *File* menu. In the *Open* dialog box, select the *Exclusive* check box (iš naujo paleiskite duomenų bazės programą: *File* → *Open Database* → *Exclusive*). Bakstelėkite OK ir iš naujo atverkite duomenų bazę: *File* → *Open Database* → dialogo langas *Open* → *DB_EXPENSES.mdb*. → jungiklis *Exclusive* → *Open* → slaptažodis *expenses* → nuspaukite *Shift* → OK ir vėl bandykite atsisakyti slaptažodžio. Jeigu viską vykdėte nuosekliai – pavyko. Patikrinkite ar dabar galėsite įkelti duomenų bazę be slaptažodžio. Nepamirškite nuspausti *Shift*.

Trečias būdas. Duomenų šifravimas - tai patikimesnė duomenų apsauga.

Uždarykite duomenų bazės langą ir iš komandų sąrašo nurodykite: *Tools* → *Security* → *Encrypt/Decrypt Database...* → dialogo langas *Encrypt/Decrypt* ir t.t. Dabar šio apsaugos būdo *Encrypt/Decrypt*, šifruoti/iššifruoti, išsamiau nenagrinėsime.

Ketvirtas būdas. Nustatyti atskiriems vartotojams ir vartotojų grupėms ribotas galimybes įtakoti sukurtos DB struktūrą ir duomenų valdymą bei galimybę pasinaudoti objektais ir duomenimis. Mokomajame leidinyje šis būdas nenagrinėjamas.

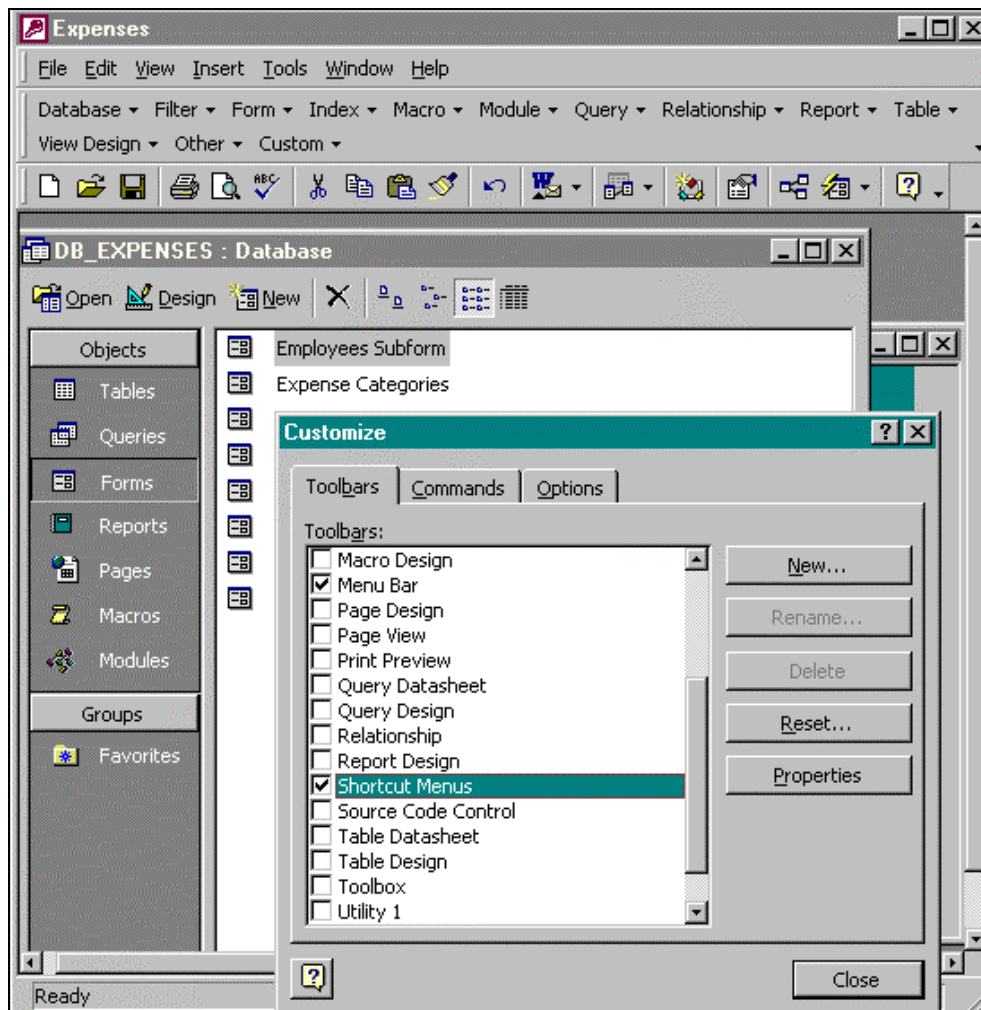
1.11. Komandų sąrašų ir mygtukų juostų parengimas

Sistema *Access* leidžia redaguoti įterptuosius komandų sąrašus, mygtukų juostas, parankinius komandų sąrašus, išvedamus pagal kontekstą, rengti savas komandas ir tokiu būdu labiau apsaugoti duomenų bazę.

1 PRATIMAS. PATYRINĖKIME ĮTERPTĄSIAS MYGTUKŲ JUOSTAS IR KOMANDAS.

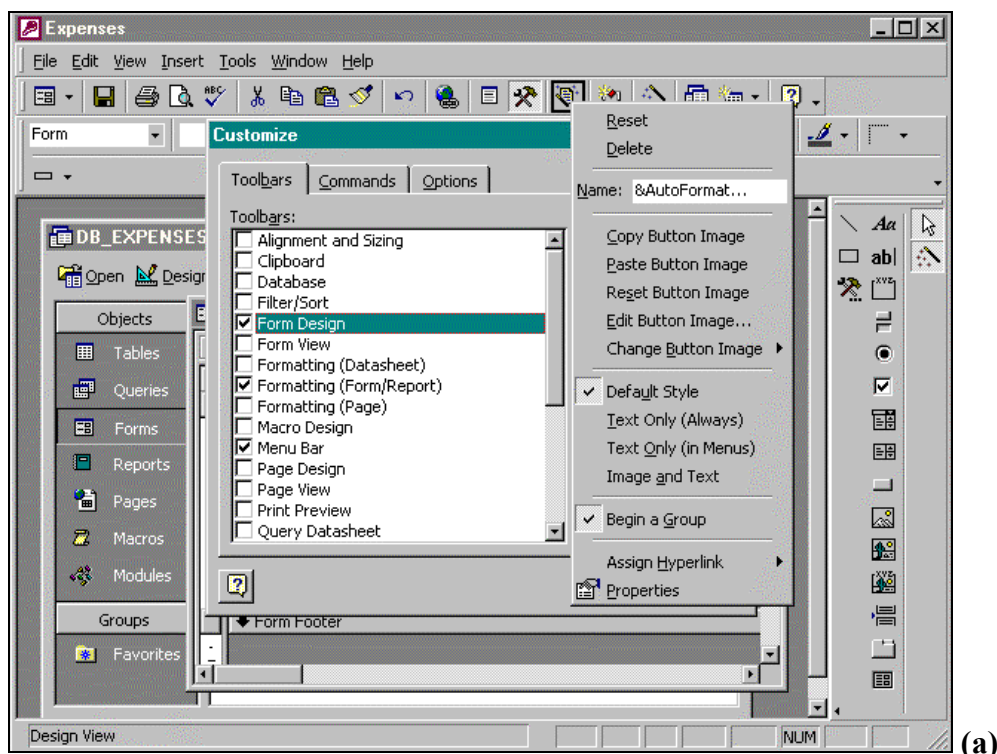
Duomenų bazės lange nustatykite pelės rodyklę mygtukų juostoje ir spustelėkite dešinį klavišą, po to nurodykite *Customize*, tinkinti. Arba: *View* → *Toolbars* → *Customize*. Ekrane matysite dialogo langą *Customize*, tinkinti, su trim kortelėmis: *Toolbars*, mygtukų juostos; *Commands*, komandos; *Options*, parinktys. Peržiūrėkite korteles, susipažinkite su pateiktaisiais parametrais.

Toolbars kortelėje suraskite *Shortcut Menus*, sparciosios komandos. Ekrane, viršuje, pamatysite sparcijų komandų sąrašų eilutę *Shortcut Menus: Database, Filter, Form* ir kt. Kiekvieną sparcijų komandų sąrašą galite išskleisti bakstelėdami prie sąrašo pavadinimo esančią išskleidžiamo sąrašo rodyklėlę „žemyn“, o prie kiekvienos komandų grupės - rodyklėlę „dešinėn“. Matysite sparcijų komandų sąrašus.

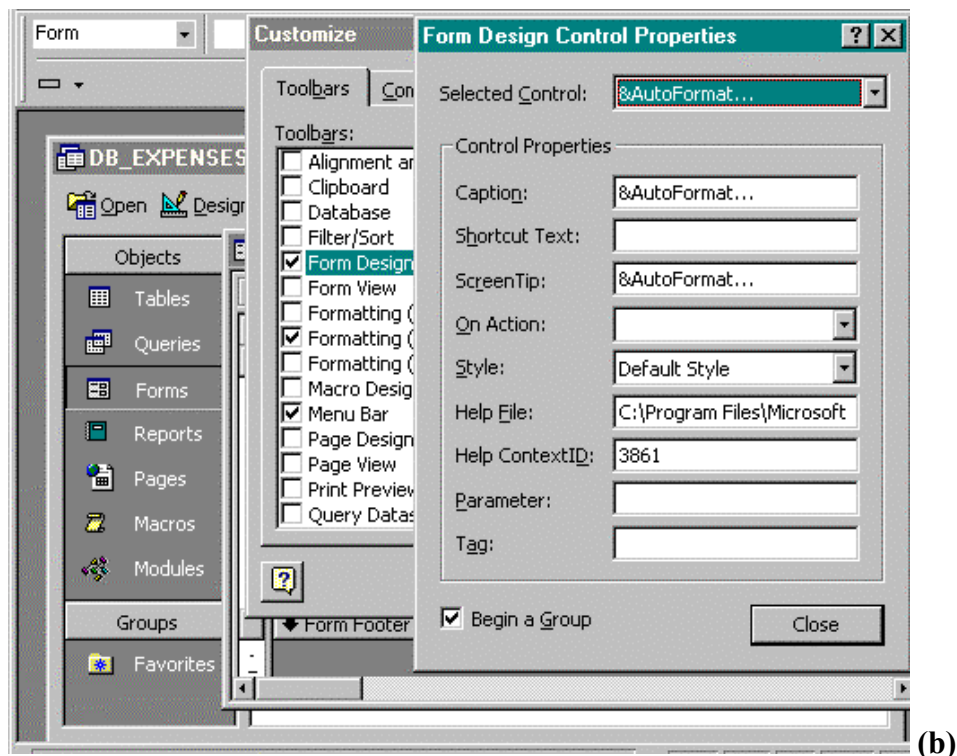


1.20 pav. Sistemos *Access* sparčiųjų komandų sąrašų, išvedamų komanda *Shortcut Menus*, peržiūra duomenų bazės *Expenses* lange

Įterptosios komandų sąrašų eilutės specializavimas. Sparčiąsias komandas komandų grupėse galima šalinti, perkelti, kopijuoti, papildyti ir t.t.



(a)



1.21 pav. DB mygtukų juostos *Form Design* mygtuko *AutoFormat* komandų sąrašas (a), atvertas pagal kontekstą (formos konstravimo lange ties pasirinktu mygtuku reikia spustelti pelės dešinį klavišą, nurodyti komandą *Customize*, vėl ties mygtuku spustelti pelės dešinį klavišą)

Pasirinkto mygtuko kaip valdomojo elemento savybių sąrašas (b) *Form Design Control Properties* išvedamas, kai komandų pagal kontekstą sąrašė pasirenkama komanda *Properties*

Sparčiųjų komandų sąrašų eilutės *Shortcut Menus* komandų sąrašai, pavyzdžiui, yra tokie: *Database*, *Filter*, *Form* ir kt.

Sparčiųjų komandų sąrašo komandų grupės, pavyzdžiui, komandų sąrašė *Database* yra: *Background*, *Form*, *Title Bar*, *Macro* ir kt.

Sparčiosios komandos komandų sąrašo *Database* grupėje *Form* yra: *Open*, *Design View*, *Print*, *Print Preview*, *Cut*, *Copy* ir kt.

- Sparčiąją komandą iš komandų grupės pašalinsite, kai nusitaisysite pele ir nutempsite į darbo langą sritį.
- Perkelsite į kitą vietą komandų grupėje arba į kitą komandų grupę tempdami į kitą poziciją.
- Kopijuosite į kitą komandų grupę, kai nuspausite klavišą *Ctrl* ir tempsite į numatytą vietą.
- Papildysite, kai kopijuosite ir įterpsite iš kitos komandų grupės, arba iš sąrašo *Commands*, tempsite į norimą papildymo vietą.
- Komandą, kuri atveria lentelę, užklausą, formą ar ataskaitą galėsite įterpti, kai rasite kortelės *Commands* sąrašė *Categories* pasirenkamą tipą, ir tempsite į reikiamą grupę. Programos vykdymui komandą yra sukurta panaudojant makrokomandą. Sąrašė *Categories* pažymėsite *All Macros*, pažymėsite reikiamą makrokomandą ir tempkite į komandų sąrašų eilutės ir grupės tinkamą vietą.
- Pakeitimus, kuriuos padarėte ir norite jų atsisakyti, sistema *Access* pašalins, kai pažymėsite komandų sąrašą arba komandų grupę ir *Customize* lange bakstelsite mygtuką *Reset*, grįžtis. Ekrane pamatysite pranešimą: *Are you sure you want to reset the changes made to the 'Shortcut Menus' toolbar?* (Ar reikia šalinti padarytus pakeitimus?)

Bakstelkite *OK*. Jeigu eksperimentavote su komandų sąrašų *Database* - pažymėkite, *Reset* ir *OK*.

- Spustelėję pelės dešinįjį klavišą galite pakeisti sparčiojo komandos mygtuko vaizdą, pavadinimą arba komandos pavadinimą (1.21 pav). Be to, galima atsisakyti komandos pakeitimų, šalinti, keisti komandą, kopijuoti, įterpti arba modifikuoti mygtuko vaizdą, komandą nurodyti tekstu arba vaizdu, panaudoti nustatytą stilių, komandų sąrašą papildyti (suskirstyti) į dalis linijomis, komanda susieti hipersaitu. Kai komandą pažymėtą *Customize* lange, reikia bakstelėti mygtuką *Properties*. Tada pasirodys savybių keitimo langas.

Specializuotos komandų sąrašų/meniu eilutės parengimas. Specializuotą komandų sąrašų/meniu eilutę galima rengti individualią - pritaikytą vienam duomenų bazės objektui, pvz., formai arba ataskaitai. Toks komandų sąrašas išvedamas, kai atveriamas objektas. Kitu atveju, specializuota - visuotinė komandų sąrašų eilutė gali būti rengiama ir pritaikoma visai duomenų banzei. Ji bus matoma visuose duomenų bazės objektų languose.

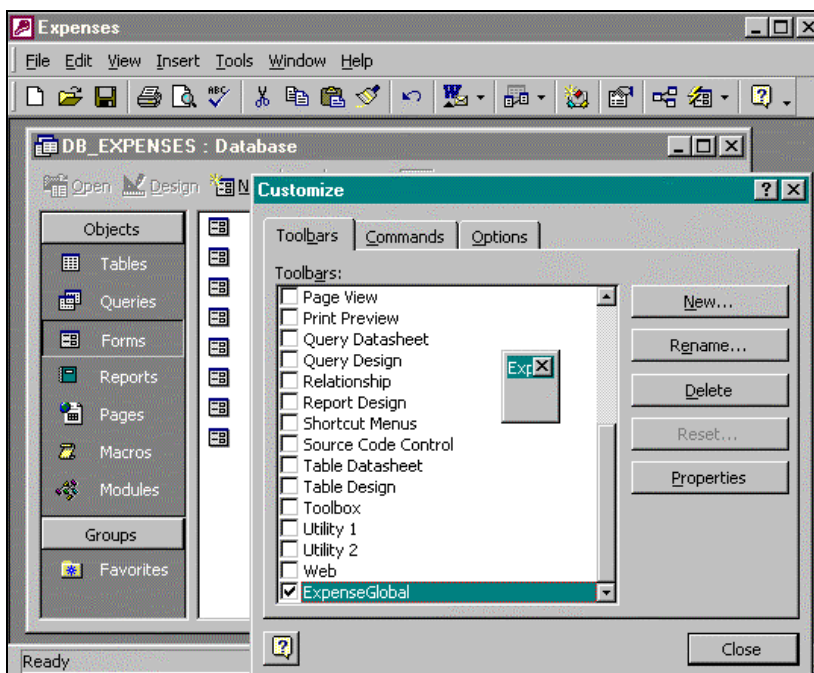
2 PRATIMAS. PARENKIME VISAI DUOMENŲ BAZEI TAIKOMĄ, T.Y. VISUOTINĘ KOMANDŲ SĄRAŠŲ EILUTĘ/JUOSTĄ:

1. Duomenų bazės lange *DB_EXPENSES:Database* nurodykite: *View* → *Toolbars* → *Customize*.

2. Dialogo lange *Customize* bakstelėkite mygtuką *New* ir lango *New Toolbar* įvesties laukelyje *Toolbar name* surinkite visuotinės komandų sąrašų juostos pavadinimą *ExpenseGlobal* ir bakstelėkite *OK*. Sistema *Access* sukurs nedidelę dar tuščią komandų sąrašų eilutę, tiksliau - dar tik komandų sąrašų eilutės pradžią *Ex* ir kortelę *Toolbars* papildys nauju dėmeniu - *ExpenseGlobal*. Ji jau pažymėta. Dabar dar tuščią naują komandų sąrašų eilutės pradžią tempkite aukštyr ir palikite žemiau mygtukų juostos *Database*.

3. Lange *Customize* bakstelėkite savybių komandos mygtuką, *Properties*. Dialogo lango *Toolbar Properties* išskleidžiamajame jame sąraše *Type*, galima nurodyti komandų eilutės tipą: *Menu Bar*, komandų sąrašas; *Toolbar*, mygtukų juosta; *Popup*, parankinių/pagal kontekstą komandų sąrašas. Nurodykite *Menu Bar* ir bakstelėkite *Close*.

4. Mygtukų juostoje *Database* komandų sąrašą *File* reikia perkelti į naują tuščią komandų sąrašų juostą. Nuspauskite klavišą *Shift* ir vilkite komandų sąrašą *File* į naują tuščią komandų sąrašų juostą. *Customize* langas turi būti atviras.



1.22 pav. Naujos mygtukų juostos *ExpenseGlobal* kūrimo pradžia lange *Customize*

5. Bakstelėkite *File* naujoje juostoje. Naujame komandų sąrašė reikia palikti tik komandas *Close*, *Save As*, *Page Setup*, *Print Preview*, *Print* ir *Exit*. Kai langas *Customize* atviras ir *Toolbar* kortelėje pažymėta *ExpenseGlobal*, bakstelėkite *File* komandų sąrašą naujoje komandų sąrašų juostoje, pažymėkite po vieną nereikalingą komandą ir tempkite iš komandų sąrašo į *Customize* langą arba į duomenų bazės langą ir pameskite.

6. Į naują komandų sąrašų juostą kopijuokite kitą komandų sąrašą *Edit*, panašiai, kaip kopijavote *File*: nusitaikykite į *Edit*, tik dabar nuspauskite klavišą *Ctrl* ir tempkite komandų sąrašą *Edit* į naują juostą, palikite dešiniau *File*. Naujame *Edit* komandų sąrašė palikite tik komandas *Undo*, *Cut*, *Copy* ir *Paste*.

7. Lango *Customize* kortelėje *Commands*, sąrašė *Categories*, pažymėkite *New Menu*. Iš kortelės *Commands* sąrašo *Commands*, esančio dešiniau *Categories*, tempkite *New Menu* į naują komandų sąrašų juostą ir palikite dešiniau *Edit*. Ties perkeltu *New Menu* spustelėkite dešinįjį pelės klavišą ir pakeiskite pavadinimą: *Name* įvesties laukelyje vietoj *New Menu* surinkite *Forms*, spustelėkite *Enter*. Kai dabar bakstelėsite naują komandų sąrašą *Forms*, matysite žemyn išskleidžiamojo sąrašo pradžią. Išskleiskite sąrašą.

8. *Customize* lango kortelėje *Commands* pažymėkite *All Forms*. Iš sąrašo *Commands* kopijuokite (vilkite) po vieną formas: *Expense Reports by Employee* ir *Expense Categories* į komandų sąrašo *Forms* išskleidžiamojo sąrašo vietą (žemiau įterptos formos vardo). Taip suformuosite *Forms* išskleidžiamąjį sąrašą. Bakstelėkite *Forms*, uždarykite sąrašą.

9. Naują komandų sąrašų eilę galite ekrane padėti bet kurioje vietoje, geriausia, kaip įprasta ją reikia šlieti prie nuolat esančios komandų mygtukų juostos.

10. Lango *Customize* kortelėje *Toolbars* pažymėkite *ExpenseGlobal* ir bakstelėkite mygtuką *Properties*. Dialogo lange *Toolbar Properties* galima leisti arba uždrausti pakeitimus naujame komandų sąrašė. Pažymėkite lauke *Selected Toolbar* pavadinimą *ExpenseGlobal* ir išjunkite nustatomojus jungiklius - taip uždrausite pakeitimus, bakstelėkite *Close*.

Tarkime, kad visuotinė komandų sąrašų eilutė jau suformuota. Ji bus išvedama į ekraną, kai nurodysite *Startup* lange atitinkamą parametą. Uždarykite langą *Customize*.

11. *Tools* → *Startup*. Dialogo lange *Startup* išskleiskite sąrašą *Menu Bar*, nurodykite *ExpenseGlobal*, bakstelėkite *OK*. Kai kitą kartą ir vėliau, kai tik paleisite veikti taikomąją programą - *Expenses*, naujas komandų sąrašas vietoj ankstesnio bus išvedamas duomenų bazės lange.

Formų ir ataskaitų individualios komandų sąrašų eilutės rengiamos labai panašiai. Skiriasi tik parengtos eilutės išvedimo nustatymas. *Startup* lange reikia nurodyti pavadinimą ir nustatyti ne parametą, bet savybę *Menu Bar*.

Individualių komandų sąrašo eilutė vaizduojama tik savo paskirties lange.

Ankstesnią komandų sąrašą atkursite, kai nuspaudę *Shift* iš naujo įkrausite duomenų bazę, kopijuosite į ankstesnę komandų eilutę *File*, grąžinsite DB komandų sąrašą *Menu Bar* bakstelėdami *Toolbars* kortelėje *Reset*. Norėdami redaguoti mygtukų juostos komandas turėsite *Toolbar Properties* lange įjungti išjungtus jungiklius.

SPARČIŲJŲ KOMANDŲ SĄRAŠŲ/MENIU RENGIMAS.

Sparčiųjų komandų sąrašų eilutės/juostos taip pat gali būti individualios kiekvienai formai/ataskaitai arba visuotinės išvedamos tuo atveju, jei forma/ataskaita neturi tokio tipo individualių komandų eilučių.

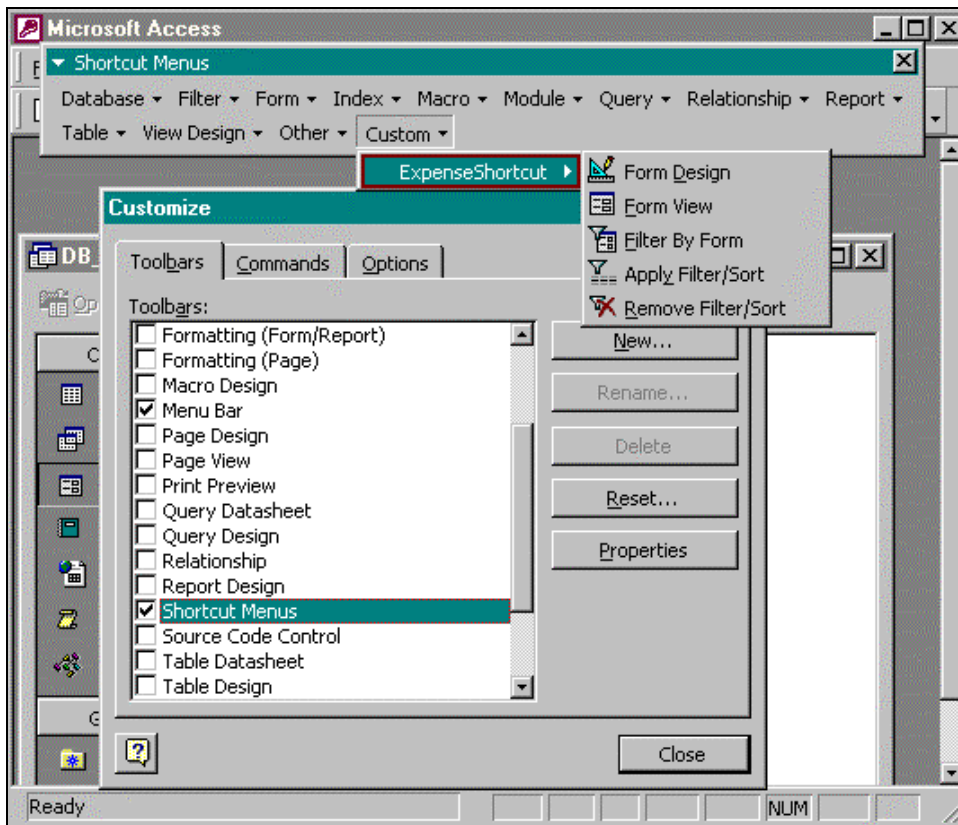
3 PRATIMAS.

Parenkime paprastą specializuotą visuotinio naudojimo sparčiųjų komandų sąrašą/meniu.

1. Duomenų bazės pagrindinėje komandų sąrašų eilutėje (ten kur *File*, *Edit*, *View* ir kt.) fiksuokite/nustatykite pelės rodyklę ir spustelėkite dešinįjį klavišą. Nurodykite *Customize*, tinkinti.
2. Dialogo lange *Customize*, kai atvira kortelė *Toolbars*, bakstelėkite mygtuką *New*. Surinkite įvesties laukelyje *Toolbar name* pavadinimą *ExpenseShortcut* ir bakstelėkite *OK*. Bakstelėkite mygtuką *Properties*. Išskleiskite sąrašą eilutėje *Type*. Ekrane gali pasirodyti pranešimas: *You have set ExpenseShortcut's Type property to Popup, which changes the toolbar to a shortcut menu* (kai nustatėte savybę *Popup* įprastinių komandų sąrašas keičiamas sparčiųjų komandų sąrašu). *The shortcut menu disappears because Expenses adds ExpenseShortcut to the Shortcut Menus Toolbar. To complete the shortcut menu, close the Toolbar Properties sheet, display the Shortcut menus toolbar, click the Custom category, and then add the commands you want* (komandų sąrašo pradžios skydelis dingsta, nes programa *Expenses* jį pertvarko į kitą tipą; galėsite tęsti, kai uždarysite savybių langą *Toolbar Properties*, aktyvinsite *Shortcut Menus* kortelėje *Toolbar*, bakstelėsite *Category*; tada galėsite rinkti tinkamas komandas). Taip ir padarykite. Pranešimo lange bakstelėkite *OK*, savybių lange *Toolbar Properties* bakstelėkite *Close*.
3. Lango *Customize*, kortelės *Toolbars*, pabaigoje pažymėkite *Shortcut Menus*. Ekrane matysite komandų sąrašų naują eilutę *Shortcut Menus: Database, Filter, Form, Index* ir t.t. Toje eilutėje suraskite *Custom* ir bakstelėkite už jo mažą rodyklėlę-trikampėlį. Bakstelėkite tokią pat rodyklėlę, esančią už pasirodžiusio *ExpenseShortcut*. Pasirodo išskleidžiamojo sąrašo pradžia. Galima kopijavimo būdu formuoti naują sparčiųjų komandų sąrašą. Tarkime, kad reikia iš komandų sąrašo *Form View* suformuoti duomenų filtravimo sparčiųjų komandų sąrašą.
4. *Shortcut Menus* eilutėje, suraskite *Form* ir bakstelėkite rodyklėlę už jo. Išskleistame komandų sąraše bakstelėkite *Datasheet View Title Bar*, po to dešinėje rodyklėlę. Matysite duomenų filtravimui būdingų (ir kt.) sparčiųjų komandų sąrašą. Tempkite po vieną sparčiąsias komandas į *ExpenseShortcut* sąrašą. Nuspauskite klavišą *Ctrl* ir tempkite iš karto į rodyklės vietą už *Custom*, o kai pasirodys *ExpenseShortcut* į rodyklėlę ir į išskleidžiamojo sąrašo vietą. Atitemptus sparčiųjų komandų mygtukus išdėstysite vieną po kito vertikaliai, t.y. *Popup*, sluoksniavimo, būdu.
5. Uždarykite langą *Customize*, kartu ir formuojama sparčiųjų komandų eilutę *Shortcut Menu*. Bakstelėkite *Close*. Dar reikia nurodyti visuotinį veikimą.
6. *Tools* → *Startup* → dialogo langas *Startup* → skirsnis *Shortcut Menu Bar*. Išsiskleiskite sąrašą ir nurodykite *ExpenseShortcut*. Bakstelėkite *OK*.

Kai kitą kartą paleisite taikomąją programą *Expenses*, suformuotas specialus sparčiųjų komandų sąrašas galės būti išvestas formoje, kai spustelėsite pelės dešinįjį klavišą.

Kitu atveju, kai norėsite matyti *ExpenseShortcut* komandų eilutę ekrane nurodykite: *View* → *Toolbars* → *Customize* → dialogo langas *Customize* → kortelė *Toolbars* → *Shortcut Menus*. Po to *Custom* rodyklėlė → *ExpenseShortcut* rodyklėlė, ir matysite vėl sparčiąsias komandas *Filter*.



1.23 pav. Sparčiųjų komandų sąrašo *ExpenseShortcut* rengimo informacinis fragmentas

Specializuotų mygtukų juostų rengimas. Vienu metu ekrane prie bet kurio lango būna tik viena komandų sąrašų eilutė ir vienas parankinis komandų sąrašas. Gali būti neviena mygtukų juosta. Tačiau *Access* turi galimybę ekrane vaizduoti iš karto keletą įterptinių arba specializuotų komandų eilučių.

4 PRATIMAS. PARENKITE VISUOTINĘ MYGTUKŲ JUOSTĄ:

1. Spustelėkite pelės dešinįjį klavišą, kai pelės rodyklė ties komandų sąrašu arba ties mygtukų juosta *Database*, ir nurodykite *Customize*.
2. Lange *Customize*, kai atvira kortelė *Toolbars*, bakstelėkite mygtuką *New*. Pavadinkite mygtukų juostą *ExpenseToolbar* ir bakstelėkite *OK*. Sistema *Access* papildys sąrašą *Toolbars* nauju pavadinimu *ExpenseToolbar* ir parodys naujos mygtukų juostos užuomazgą *Ex*.
3. Atverskite kortelę *Commands*. Sąrašė *Categories* bakstelėkite kategoriją *Edit* ir kopijuokite (nuspauskite *Ctrl* ir vilkite) į naują mygtukų juostą komandas: *Undo*, *Cut*, *Copy* ir *Paste*. Iš kategorijos *Records* kopijuokite *Sort Ascending* ir *Filter by Form*.
4. Nuvilkite suformuotą naują mygtukų juostą ir palikite ją tarp komandų sąrašų eilutės ir mygtukų juostos *Database*.
5. *Customize* lango kortelėje *Toolbars* pažymėkite naujos mygtukų juostos pavadinimą *ExpenseToolbar* ir bakstelėkite savybių mygtuką *Properties*. Dialogo lango *Toolbar Properties* įvesties lauke *Selected Toolbar* pažymėkite *ExpenseToolbar*, išjunkite žemiau esančius įjungtus jungiklius, taip uždrausite įvairius pakeitimus ir bakstelėkite *Close*.
6. Lango *Customize* sąrašė palikite parengtus tik *Menu Bar* ir *ExpenseToolbar*. Pašalinkite *ExpenseGlobal*. Pažymėkite ir bakstelėkite mygtuką *Delete*, po to pranešimo lange bakstelėkite *OK*.

1.12. Access duomenų bazės automatizavimo rezultatų apžvalga

Nagrinėdami taikomąją programą *Expenses*, susipažinote su kai kuriomis priemonėmis, matėte, kaip netaikant programavimo galima automatizuoti duomenų bazių objektų bei elementų valdymą.

PRATIMAS. DAR KARTĄ APŽVELKITE IR ĮSIDĖMĖKITE KĄ PAVYKO NUSTATYTI:

1. Paleiskite iš naujo veikti sistemą *Access*, atsisakykite papildomojo lango *Microsoft Access*: bakstelėkite mygtuką *Cancel*. Duomenų bazę *DB_EXPENSES* įkraukite iš komandų sąrašo *File*.

2. Dialogo lango *Password Required* įvesties laukelyje *Enter database password* surinkite slaptažodį *expenses* ir bakstelėkite *OK*. Slaptažodį visuomet reikia surinkti tokiais simboliais, kokiais jis buvo įvestas pirmą kartą. Didžiosios ir mažosios raidės slaptažodyje suprantamos kaip skirtingi simboliai. Pvz., netiks *Expenses*.

Ekrane matysite taikomosios programos lange *Expenses* komutacijos skydelio langą *Switchboard* su visuotiniu komandų sąrašu *ExpenseGlobal* ir specializuotą mygtukų juostą Duomenų bazės lango ekrane dabar nematysite. Jeigu išvestas ne pagrindinis valdymo langas, bakstelėkite mygtuką *Return to Main Switchboard*. Pagrindiniame komutacijos skydelio lange *Main Switchboard* bakstelėkite *Enter/View Expense Reports by Employee*. Formoje *Expense Reports by Employee* spustelėkite pelės dešinįjį klavišą, ekrane matysite sparčiųjų komandų sąrašą. Be to, formoje turite ir kitus įrengtus komandų mygtukus. Uždarykite formą: formos lange bakstelėkite grįžties mygtuką *Return*. Darbą baigsite su taikomąja programa *Expenses*, kai lange *Main Switchboard Expenses*, bakstelėsite mygtuką *Exit this database*.

Jeigu sistemos *Access* lange yra nustatyta mygtukų juosta *Web*, pasaulinis voratinklis, o jei nėra nustatykite: *View* → *Toolbars* → *Customize* → kortelė *Toolbar* → *Web*. Joje matysite adresų išskleidžiamąjį sąrašą, jų tarpe ir taikomosios programos *Expenses* adresą, pvz.: *C:\My Documents\Macro&VBA\DB_EXPENSES.mdb*. Pažymėkite adresą ir spustelėkite *Enter*. Taip vėl galėsite paleisti taikomąją programą *Expenses*. Tik dabar, kai surinksite slaptažodį *expenses*, nuspauskite klavišą *Shift* ir bakstelėkite *OK*. Dabar neveikia anksčiau nustatyta kreipties į duomenų bazę apsauga ir ekrane matysite duomenų bazės langą *DB_EXPENSES:Database*. Kai duomenų bazės objektų kortelėje *Forms* pažymėsite formą *Switchboard* ir bakstelėsite komandos mygtuką *Open*, matysite *Main Switchboard* langą. Kai nurodysite komandą: *Window* → *Tile Vertically*, matysite ekrane abu langus.

3. Baikite darbą. Uždarykite visus langus.

Panaudotus duomenų bazės taikomosios programos automatizavimo būdus galima taikyti žymiai plačiau. Pavyzdžiui, paslaugų programa *Command Button Wizard* leidžia supaprastinti dvikryptį judėjimą į visas formas ir ataskaitas, be to, sudaro galimybę automatizuoti kitas užduotis, o *Combo Box Wizard* – įgalina panaudoti kombinuotus blokus *Lookup Switchboard Manager* leidžia įterpti naujus objektus į formą *Switchboard*. Sistema *Access* taip pat suteikia galimybę įrengti papildomų komandų sąrašų/meniu eilutes, sparčiųjų komandų mygtukų darinius, palengvinančius darbą su taikomąja programa.

Galima teigti, kad panaudojant vien tik vedlių - *Database Wizards* paslaugų programas ir įvairius savybių nustatymo langus su *Startup* ir *Customize* galima gana greitai automatizuoti duomenų bazę (duomenų bazės valdymo taikomąją programą). Tačiau *Database Wizard* galima panaudoti efektyviai tik pagal sistemos *Access* šablonus rengiamoms duomenų bazėms. Sistemos *Access* šablonų skaičius yra ribotas. Jus galite matyti iš naujo paleisdami programą *Microsoft Access*, kai bakstelėsite mygtuką *New Database* ir dialogo lange *New* atversite kortelę *Databases*.

1.13 UŽDUOTIS.

1. Įkrauti atsisiųsta DB *Expenses.mdb* (slaptažodis *expenses*) ir nuosekliai peržiūrėti, patikrinti veikimą.
2. Įkrauti konvertuotą DB *Konvert-Expenses*, peržiūrėti, patikrinti veikimą.
3. Pastebėti DB struktūros ir veikimo ypatumus peržiūrėtose duomenų bazėse ir palyginti su duomenų baze *DB_EXPENSES*, su kuria atlikote pratimus.

1.14 UŽDUOTIS.

1. Tęsti ir baigti anksčiau pradėtos su lietuviškais objektų vardais duomenų bazės *DB_SĄNAUDOS* rengimą.
2. Išbandyti veikimą.

1.15 UŽDUOTIS.

1. Parengti savo iniciatyva naują duomenų bazę, kurios sudėtyje būtų nors 5 lentelės, trys formos (viena iš jų sudėtinė), dvi ataskaitos ir kiti objektai bei elementai apipavidalinti ir pateikti panašiai kaip nagrinėtose duomenų bazėse. Objektus galima kopijuoti iš kitų duomenų bazių ir atitinkamai suredaguoti.
2. Pademonstruoti objektus ir duomenų bazės panaudojimo galimybes.
3. Įsidėmėti, mokėti aiškinti ir parodyti duomenų bazės sukūrimo ir objektų rengimo bei redagavimo atskirus momentus.

1.13. Programavimo taikymas duomenų bazėms automatizuoti

Rengdami kitas *Access* duomenų bases (ne pagal *Access* šablonus) *Database Wizard* pagalbos ir kitų pagalbinių priemonių efektyviai panaudoti negalėsite. Pavyzdžiui, negalima patikrinti ir patvirtinti duomenų atitikties ir tinkamumo (*matching*), negalima vykdyti logiškai susietų veiksmų sekų sistemos būsenai pakeisti, kurios vadinamos - t r a n s a k c i j o m i s (*transaction*). Be to, visiškai netinka, kai kartu su *Access* panaudojami kitų programų dokumentai, pvz., *Word*, *Excel*.

Sukurtas savas duomenų bases galima automatizuoti taikant programavimą.

***Access* duomenų bazėms programuoti taikomas programavimas *VBA* - *Visual Basic for Applications* Makrokomandų rengimas kartais vadinamas makroprogramavimu, tačiau tai nėra tiesioginis programavimas. Makrokomandų procedūras *Access* sukuria automatiškai.** Abu būdai yra visiškai savarankiški ir nepriklauso vienas nuo kito. Tačiau programuojant makrokalba ir *VBA* kalba panaudojami tie patys standartinių instrukcijų rinkiniai, vykdančys tipinius veiksmus, pvz., atverti/užverti, t.y. atidaryti/uždaryti objektus ar įkelti duomenis. Tokios standartinės instrukcijos makroprogramavimo atveju vadinamos v e i k s m a i s , o *VBA* - m e t o d a i s .

Makrokomandas parengti yra žymiai paprasčiau, tačiau, kai norima aprėpti sudėtingesnius atvejus, pvz., valdyti sistemos *Access* reakcijas į nenumatytas klaidas ir kt. be *VBA* procedūrų neapsieisite. Makrokomandos gali būti naudojamos *VBA* programose klaviatūros komandoms parengti.

- Makrokalba paprastesnė negu *VBA*, ją lengviau įsisavinti, tačiau jos galimybės ribotos.
- Kai kada makrokomandos veikia greičiau, todėl daugelis formų ir ataskaitų dažnai projektuojamos panaudojant makrokomandas.
- Makrokomandas galima visuomet pakeisti *VBA* procedūromis, todėl *VBA* programos rengimą paprasčiau pradėti nuo makrokomandų.

1.14. Duomenų valdymo kalbos

Be programavimo kalbų - makrokalbos ir *VBA* programuotojas turi žinoti, kad yra dar dvi duomenų valdymo kalbos: *SQL* - *Structured Query Language*, struktūrinė užklausų kalba, ir *DAO* - *Data Access Objects*, kreiptis į objektų duomenis.

Duomenų valdymo kalbos - nėra autonominės programavimo kalbos. Jos atlieka tik kreipties į duomenis organizavimo vaidmenį bei kartu su programavimo kalbomis ženklina iš lentelių išrenkamus arba į lenteles įkeliamus duomenis.

Microsoft Access dalis su kuria dirba vartotojas kurdamas ir išvesdamas duomenų bazės objektus yra vadinama *Access Application*.

Sistemos *Access* dalis, naudojama duomenų bazės objektams valdyti, vadinama *duomenų bazės branduoliu* (*database engine*). *Access* duomenų bazėse duomenų bazės branduolys vadinamas *Jet*, o *Access* projektuose – *Microsoft Database Engine (MSDE)*. Atskiros *Access* dalys sąveikauja duomenų valdymo kalbomis (*data-access language*).

SQL kalba konstruojamos užklausos. Be to, *SQL* dažnai panaudojama *Form*, *Report*, *Combo Box* ir *List Box Wizard* paslaugų programose, taip pat ir makrokomandose bei *VBA* programose lentelių ir užklausų duomenims organizuoti.

DAO taikoma tik *VBA* programose duomenų bazės branduolio valdomiems objektams tvarkyti, instrukcijoms rengti. Ženklina duomenis ir sudaro galimybę rengti ir modifikuoti duomenų bazių objektus.

Be to, dar yra kalba *ADO* – *ActiveX Data Objects* – tai interpretatorius tarp duomenų bazės objekto *Application* ir duomenų bazės dalies, vadinamos branduoliu.

1.15. Santrauka

Pirmajame skyriuje buvo analizuojamos sistemos *Access* priemonės naudojamos duomenų bazėms automatizuoti be programavimo.

- Paslaugų programa *Dabase Wizard* sukuria iš dalies automatizuotas taikomas programas, papildomas komutacijos formomis *Switchboard*, komandų mygtukais, kurie gali atverti ir sinchronizuoti formas ir ataskaitas, bei papildomas specialios paskirties dialogų langais vartotojo informacijai įvesti.
- Programa - vedlys *Command Button Wizard* gali įrengti komandų mygtukus su daugiau nei 30 operacijų automatizavimo scenarijais.
- Programa - vedlys *Combo Box Wizard* leidžia rengti kombinuotus paieškos blokus reikiams įrašams surasti ir vaizduoti.
- *Switchboard Manager* parengia ir modifikuoja duomenų bazės komutacijos formą *Switchboard*.
- Dialogo langas *Startup* paleidžia programas ir užtikrina apsaugą.
- Dialogo langas *Customize* sudaro galimybę sudaryti ir atitinkamai suderinti komandų sąrašus, sparčiųjų komandų ir mygtukų eilutes.
- *Access* programavimo aplinka yra valdoma įvykiais. Valdomas elementas, forma arba ataskaita, priklausomai nuo vartotojo poveikio, atpažįsta pokyčius, vadinamus įvykiais. Sistema *Access*, atsižvelgdama į įvykius vykdo programas.

Access sistemoje panaudojamos dvi programavimo kalbos: makrokomandų programavimo ir taikomųjų programų programavimo - *Visual Basic for Applications (VBA)*. Makrokomandų programavimo metu parengiami makrokomandų objektai *Macros*. *VBA* programavimo metu parengiamos procedūros, kurios sudaro standartinius modulius *Modules* arba įterpiamuosius į objektus formų ir ataskaitų

2. Objektai ir įvykiai

- Objektų samprata, vardų sudarymas
- Objektų apibūdinimas nustatytosiomis savybėmis
- Aplinkos parametrų, duomenų bazių savybių nustatymas
- Veiksmai ir įvykiai su objektais
- *Access* įvykių modeliai
- Nustatytosios standartinės elgsenos keitimas

Šiame skyriuje iš esmės nagrinėjamos dvi fundamentinės makroprogramavimo ir *VBA* programavimo sąvokos - o b j e k t a i ir į v y k i a i .

2.1. Objektinis mąstymo būdas

Net ir paprasčiausią duomenų *Access* bazę sudaro gausybė nevienareikšmių sudėtinių dalių - dėmenų/elementų, tai: lentelės, laukai, indeksai, sąlygos, užklausos, formos, įvairūs valdomieji elementai, ataskaitos, savybės ir t.t.

Valdomieji elementai *controls* - tai duomenų formų arba ataskaitų lango darbo srities elementų/dėmenų pavadinimai, tekstai, laukai - jų reikšmės ir pavadinimai, mygtukai, jungikliai, piešiniai, diagramos, nubrėžtos linijos ir kt. *V a l d o m a s i s e l e m e n t a s* gali būti pats valdomas (taip esti dažniausiai) arba vykdyti valdymo funkciją. Pavyzdžiui, duomenų formoje pažymėtą valdomąjį elementą - lauką galima tempti į kitą darbo lango vietą, t. y. lauką galima valdyti. Kitu atveju formos darbo lango srityje įrengtą valdomąjį elementą - formos uždarymo mygtuką galima ne tik valdyti, t. y. jį patį perkelti į kitą vietą, bet tuo mygtuku nurodyti valdymo komandą, pavyzdžiui, uždaryti duomenų formą.

Nuo dabar tik įprastus pagrindinius *Access* duomenų bazių objektus: *Tables, Queries, Forms, Reports, Macro, Modules*, kaip vadinome taip ir toliau vadinsime pagrindiniais objektais, o kiekvieną vartotojo arba sistemos *Access* sukuriamą bet koki kitą duomenų bazės dėmenį/elementą vadinsime tiesiog objektu. To reikalauja objektinis mąstymas.

Microsoft Access projekte galima išskirti tokius būdingus objektus: lenteles, pateiktis, išsaugomas procedūras ir diagramas, formas, puslapius, makrokomandas ir modulius. Be to, lentelės, pateiktys, išsaugomos procedūros ir diagramos saugomos serveriuose. Kiti objektai *Access* aplinkoje.

Serveris, tarnybinė stotis – *server* – kompiuteris, aptarnaujantis kompiuterių tinklus, jų ryšius bei vykdomas duomenų mainus pagal vartotojų paraiškas (serveris gali būti ir specialios paskirties taikomoji programa/prietaika)

Dabar apžvelgsime svarbiausius objektinio mąstymo aspektus.

Aspektas - požiūris, kuriuo koks nors dalykas, reiškinys tiriamas, svarstomas.

Objektas - *object* - realybė (daiktas, procesas, reiškinys), kurią galima apibūdinti sąvoka, surinkti apie ją duomenis.

O b j e k t a s - tai tam tikra, lyg svogūno tipo daugiasluoksnė struktūra. Išoriniu požiūriu objektas - yra tai, ką galima betarpiškai panaudoti arba keisti nustatant interaktyviuoju būdu kitas savybes arba vykdamas makrokomandas ir *VBA* procedūras.

Objektų vardai. Lengva suprasti, kad bendrieji objektų vardai, besiskiriantys tik numeriais, pvz., *Control1, Control2, ..., ControlN* yra mažiau informatyvūs, negu specializuoti su naudingos informacijos tarpais, pvz., *textLastName, cmdReturn, cboCustomerID*. Parenkant vardus reikia prisilaikyti vardų sudarymo standartų.

Daugelis *Access* profesionalių programuotojų pirmenybę teikia vardų sudarymo vengriškajam stiliui. Tačiau paslaugų programų meistrai *Wizards* nesilaiko vengriškojo stiliaus. Taip pat ir žemiau nagrinėjamo *Access* duomenų bazės pavyzdžio *Northwind* objektų vardai sudaryti kitaip. Tačiau projektuojant naujas duomenų bazes patartina objektų vardus sudaryti atsižvelgiant į taip vadinamo vengriškojo stiliaus rekomendacijos. Vengriškojo vardų stiliaus, turinčio keturias dalis, sintaksė:

[priešdėlis] [požymis] [pagrindinis-vardas] [priesaga]

Pagrindinis-vardas, *name*, - objekto būdingas vardas iš vieno arba dviejų žodžių, prasidedančių didžiąja raide ir rašomų be tarpo, pvz., *CustomerLabels*

Požymis, *tag*, - nurodo objekto tipą, rašomą iš trijų ar keturių mažųjų raidžių.

Priešdėlis, *prefix*, - modifikuoja požymį ir rašomas viena mažąja raide. Požymis gali turėti vieną arba du priešdėlius, papildomai nusakančius objekto tipą.

Priesaga, *suffix*, - modifikuoja pagrindinį vardą. Gali būti viena arba daugiau. Priesaga teikia papildomų žinių apie objekto panaudojimą.

Duomenų bazių pagrindinių objektų požymių tag pavyzdžiai: Tables- tbl, Query- qry, Form- frm, Report- rpt, Macro- mcr, Module- bas.

Sudėtingesni požymiai: *qapp*- papildymo užklausa – *append*; *qflt*- filtravimo užklausa – *filter*.

Valdomųjų elementų/objektų požymių tag pavyzdžiai: Check box- chk, Combo box- cbo, Command button- cmd, Image- img, Label- lbl, Line- lin, List box- lst, Text box- txt, Subreport- srp. (išsamiau žr.: Table 2.1 ir 2.2, kompaktinis diskas: \MSO-97\Lenteles\Chapter2.pdf)

Vardai VBA programose. VBA programos operuoja kintamaisiais. Kintamųjų reikšmės vykdant operacijas kinta. Rengiant VBA programą kintamasis skelbiamas/skiriamas modulio operatoriumi, o vykdant programą rezervuojama atminties ląstelė kintamojo reikšmei saugoti.

K i n t a m a s i s – tai atmintinės laikinoji sritis, kuriai suteikiamas vardas. Atmintinės laikinoji sritis naudojama kintamojo reikšmei saugoti - objektui nurodyti. Makroprogramose kintamiesiems tiesiogiai atminties rezervuoti negalima, tenka kintamuosius imituoti papildomais valdomaisiais elementais.

Priešdėlių (prefix) VBA pavyzdžiai: array- a, Currency- cur, Date/Time- dtm, Double- dbl, Integer- int, Long- lng, Memo- mem, Single- sng, Variant- var, Object- obj.

VBA programose kitokiu būdu nei makroprogramose perteikiami duomenų bazės rinkmenoje esantys duomenys.

Sistema *Microsoft Access* įdiegiama kaip dvi skirtingos paskirties dalys.

Pirmoji - t a i k o m ū j ū p r o g r a m ū d a l i s *Access Application* - turi visas priemones įprastam *Access* darbui taip pat ir priemones makrokomandoms ir programoms VBA rengti ir vykdyti.

Antroji dalis - p r o g r a m o s b r a n d u o l y s *Jet* - turi visas priemones duomenų bazei valdyti ir duomenis saugoti. Vykdam makrokomandas abi dalys veikia automatiškai. Atskirai kreiptis į *Jet* dalį galima tik VBA programomis.

Kodiniai objektų vardai, parengtosiose taikomosiose programose taip pat ir antraštės, darbo langų pavadinimų srityse rašomi labai skirtingai, pavyzdžiui, *rptMailLabelCust - Mailing Labels:Customers*, pašto adresai: užsakovai; *cboCustomerID - Lookup Customer*, užsakovo paieška; *frmCustomers - Customers*, užsakovai.

Vardų keitimas. Vardus keisti tenka gana dažnai. *Microsoft Access* neturi automatizuotos vardų keitimo galimybės. Tad keisti objekto vardą (lentelės, lauko, valdomojo elemento, formos, užklauso, makrokomandos, modulio ir kt.) galėsite tik rankiniu būdu.

Programose VBA vardų keitimą palengvina komanda *Replace*.

2.2. Objekto savybių aprašas

Savybės charakterizuoja objekto ypatumus. Kiekvienam objektui: duomenų bazei, lentelei, formai, valdomajam elementui ir t.t. yra būdinga tam tikra savybių visuma.

Savybė - *property* - būdingiausias bruožas, ypatybė.

Dažnai tenka kalbėti tik apie dabartinę savybių visumą, nes savybes galima keisti (arba jos gali kisti). Dabartinių savybių rinkinys atmintyje yra vadinamas *objekto būseną*.

Objektams būdingos dviejų rūšių būsenos savybės:

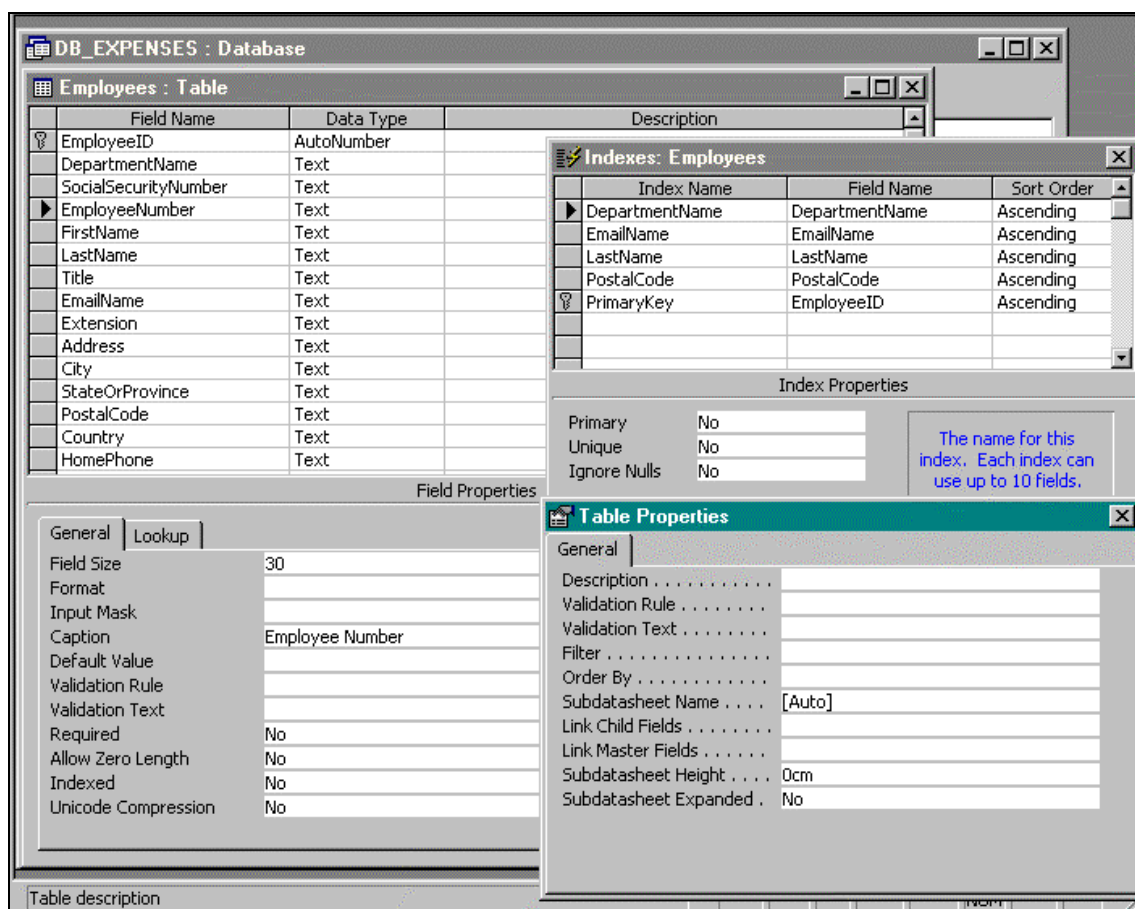
Konstravimo būsenos savybės (*design-time properties*). Savybės nustatomos konstravimo lange (pateiktis *Design*) iš objekto savybių sąrašo.

Vykdymo būsenos savybės (*run-time properties*). Savybės nustatomos arba tik nuskaitomos taikomosios programos vykdymo metu. Objekto savybių sąrašė jos nerodomas ir iš sąrašo nenurodomos. Tai vykdomo darbo pateiktys su objektais: *Datasheet*, *Form View* arba *Print Preview*, taip pat – su makrokomandomis arba VBA procedūromis.

2.3. Konstravimo būsenos savybių apžvalga

Duomenų bazės lange kiekvienas objektas turi vieną ar daugiau savybių sąrašų.

Konstravimo būsenos savybių sąrašus galima išskleisti mygtukų juostos mygtuku *Properties* arba iš komandų sąrašo *View → Properties*.



2.24 pav. Lentelės *Employees* laukų savybių *Field Properties*, laukų indeksavimo savybių *Indexes* ir lentelės savybių *Table Properties* sąrašų rodiniai

Laukų savybės pateikiamos dvejose kortelėse *General* ir *Lookup*. *General*, bendrosios savybės, kortelėje gali būti savybės: *FieldSize*, lauko dydis; *Format*, formatas; *InputMask*, įvesties kaukė; duomenų tinkamumo tikrinimo - *ValidationRule*, atitikimo tikrinimas; *Required*, būtina reikšmė ir kt. Kortelė *Lookup*, keitiniai, gali būti panaudojama paieškos/pateikties lauko savybėms nustatyti. *Lookup* palengvina duomenų įvedimą: kai lange *Datasheet* bakstelėsite lauko ląstelėje arba valdomąjį elementą, susietą su paieškos lauku, išvedamas kombinuotasis galimų reikšmių sąrašas. *Lookup* duomenų savybės yra, pvz., *RowSource*, *BoundColumn*; konstravimo savybės, pvz., *ListRows* ir kt.

Konstravimo būsenos savybės, nustatytas pagal savybių sąrašus, galima keisti makrokomandų arba *VBA* procedūrų vykdymo metu, tačiau konstravimo lange nustatytos savybės išlieka.

PRATIMAS.

Šiame skyriuje kartais reikės įkrauti duomenų bazę (DB), kuri buvo nagrinėta pirmame skyriuje – *DB_EXPENSE*. Anksčiau vartotojo iniciatyva padarytos nuostatos dabar nereikalingos. Jos dabar gali net trukdyti. Be to, net įdomu, kaip atsisakyti vartotojo padarytų nuostatų. Padarykime pratimą sudėtingesni, tardami, kad reikia išsaugoti DB su anksčiau nustatytomis vartotojo nuostatomis, sakykim, ji bus reikalinga vėliau, bet dabar tolimesniam tyrinėjimui reikia turėti tokia pat DB su įprastomis standartinėmis nuostatomis ir nepakeistu vardu.

Pasirinkime tokius veiksmus (be abejo, jie nėra vieninteliai), kurie kartu pagerintu besikaupiančią kompiuterio aplanke informaciją:

1. Kompiuteryje, savo darbo aplanke, pvz., *C:\My Documents\Macro&VBA*, kuriame dabar jau yra keletas DB (jas nagrinėjote pirmame skyriuje): *Expenses*, *DB_EXPENSES*, *Copy of DB_EXPENSES*, *DB_SANAUDOS*, *Konvert-Expenses*, sukurkite pirmo skyriaus aplanką *PIRMADALIS-1sk* ir antro skyriaus aplanką *PIRMADALIS-2sk*.
2. Perkelkite į aplanką *PIRMADALIS-1sk* visas DB, su kuriomis atlikote pirmo skyriaus pratimus, taip pat ir *DB_EXPENSES* su parengtomis vartotojo nuostatomis.
3. Kopijuokite iš aplanko *PIRMADALIS-1sk* duomenų bazę *DB_EXPENSES* su parengtomis vartotojo nuostatomis į aplanką *PIRMADALIS-2sk*.
4. Patikrinkite *DB_EXPENSES* paleidimą (su klavišu *Shift* ir be jo).
5. Dabar vykdykite 2.16 užduotį.

2.16 UŽDUOTIS.

Duomenų bazei *DB_EXPENSES* reikia grąžinti standartines nuostatas, kurios pirmo skyriaus pabaigoje buvo pakeistos vartotojo pasirinktomis: pakeistas sistemos pavadinimas, sukurti komandų sąrašų, mygtukų dariniai ir kt.

4. Įkraukite *DB_EXPENSES* iš darbo aplanko *PIRMADALIS-2sk*, laikydami nuspaustą klavišą *Shift*.
5. Gražinkite dialogo lange *Startup* (*Tools* → *Startup*) standartinius parametrus: įvesties lauke *Application Title* pakeiskite pavadinimą *Expenses* į *Microsoft Access*. Įvesties lauke *Shortcut Menu Bar* vietoj *ExpenseShortcut* nustatykite *default*, standartinis, ir *Menu Bar* vietoj *ExpenseGlobal* taip pat nustatykite *default*.
6. Išveskite mygtukų juostų sąrašą (*View* → *Toolbars* → *Customize*), šalinkite *ExpenseToolbar* (pažymėti ir bakstelti komandos mygtuką *Delete*). Kai rodomas DB pagrindinis langas, *Toolbars* sąrašė turi būti įjungtas jungiklis tik *Database* ir *Menu Bar*.
7. Uždarykite *DB_EXPENSES* ir vėl įkraukite kaip įprasta - dabar klavišo *Shift* nuspausti nebereikia.
8. Įsitikinkite, ar ekrane matote DB su įprastomis standartinėmis nuostatomis.

9. Nustatykite DB rinkmenai *DB_EXPENSES.mdb* požymį „tik skaityti“ (ties rinkmenos vardu spustelkit pelės dešinį klavišą, nurodykite *Properties*, įjunkite jungiklį *Read-only*).
10. Dar kartą paleiskite *DB_EXPENSES*, peržiūrėkite pagrindinius objektus, patikrinkite veikimą. Pastebėsite, kad, kai formoje *Expense Reports by Employee* bakstelsite mygtuką *Expense Report Form*, sistema ataskaitos dabar atversti negali, išveda pranešimą: *The command or action 'SaveRecord' isn't available now*. Kol kas palikite nustatytą požymį. Kai norėsite, arba reikės, atšauksite.

2.4. Vykdomo būsenos savybių apžvalga

Vykdomo būsenos savybės savybių sąrašuose nepateikiamos. Jos nustatomos arba nuskaitomos tik makrokomandos arba *VBA* procedūros vykdymo metu.

Vykdomo būsenos savybių pavyzdžių lentelė:

Savybė	Objektas	Aprašymas
<i>Value</i>	Valdomieji elementai	Nurodo valdomojo elemento tekstą. Nustato, kas pažymėta.
<i>Painting</i>	Formos ir ataskaitos	Nustato ar forma ar ataskaita bus kartotinai perkelta į ekraną.
<i>Text</i>	Laukas, sąrašo blokas arba jo elementas	Nustato ir išveda valdomojo elemento tekstą ar jo dalį.
<i>Bookmark</i>	Forma	Nustato formos įrašui žymelę/gairelę
<i>Selected</i>	Sąrašo blokas	Išrenka elementą arba nustato ar jis pažymėtas sąrašo bloke

Savybės *Dirty* „tik skaityti“ (pvz., ar buvo modifikuotas objektas po jo paskutinio naudojimo):

Savybė	Objektas	Aprašymas
<i>Form</i>	Forma	Nurodoma pati forma arba pagalbinė forma
<i>ActiveControl</i>	Ekranas	Nurodomas aktyvusis valdomasis elementas
<i>Count</i>	Atviros formos arba ataskaitos	Nustato atvertų objektų arba valdomųjų elementų skaičių
<i>Me</i>	Forma arba ataskaita	Nurodomas pats objektas, arba, kuris siejamas su papildomais objektais (tik <i>VBA</i>)
<i>CurrentObjectName</i>	Duomenų bazės lango objektas ar projektas <i>Project</i>	Nustato duomenų bazės lango veikiamojo objekto arba projekto lango objekto vardą
<i>CurrentRecord</i>	Forma	Atpažįsta aktyvų duomenų formos įrašą
<i>HasData</i>	Forma arba ataskaita	Nustato ar forma arba ataskaita siejama su tuščia įrašų seka

Kiti savybių sąrašai. Sistemos *Access* darbo aplinkai taikomos papildomos savybės ir charakteristikos pateikiamos formose arba dialogų languose. Daugeliu atveju tokios savybės nustatomos tik *VBA* procedūromis. Tik paleisties parametrus galima nustatyti makrokomandomis.

2.5. Paleisties parametrų nustatymas

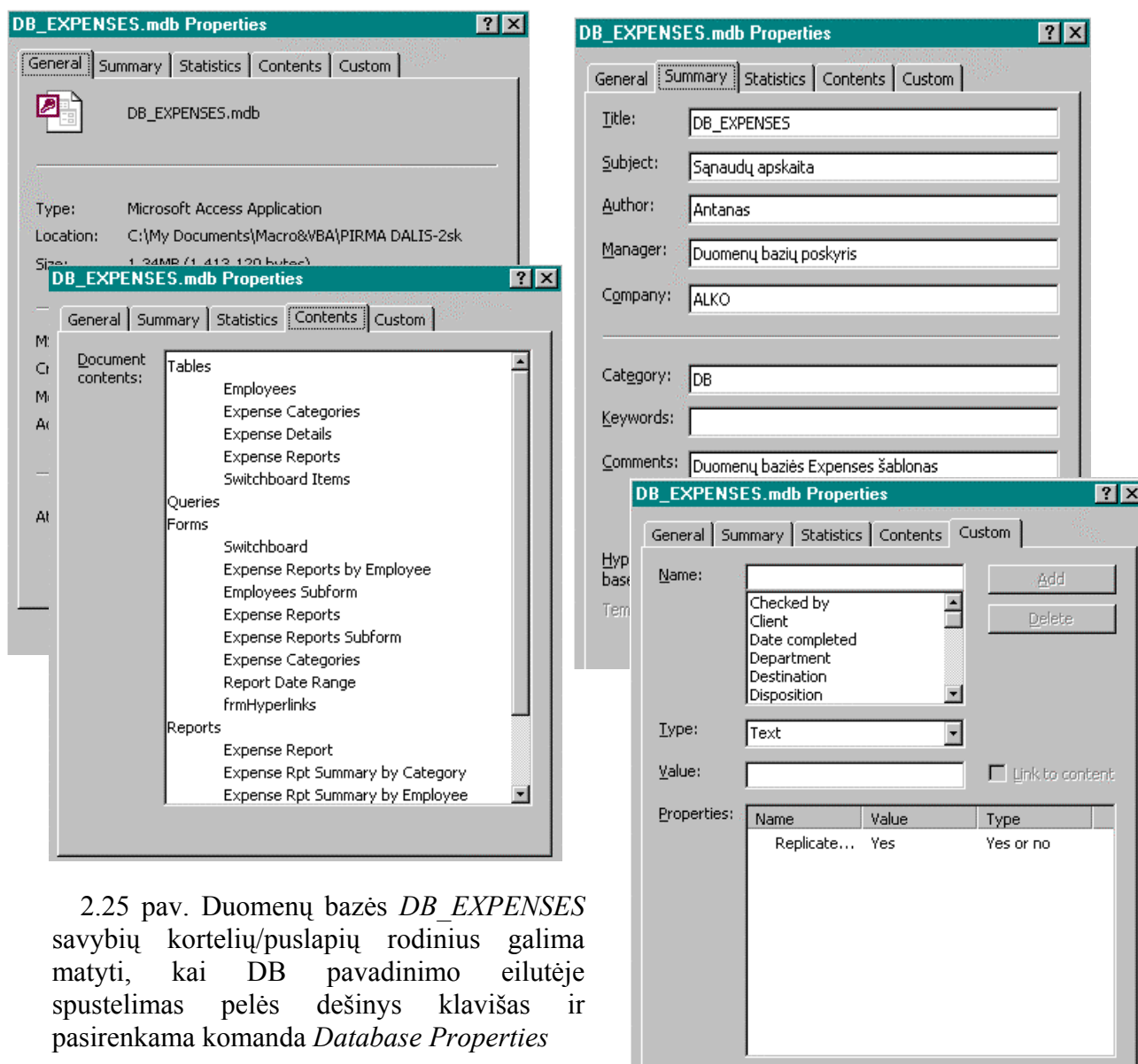
Paleisties parametrus galima nustatyti dialogo lange *Startup* (*Tools* → *Startup*). Be to, tą galima padaryti makrokomanda ir *VBA* procedūra.

Parametras - *parameter* - dydis, apibūdinantis proceso, reiškinių, sistemos savybes; objektas, su kuriuo atliekama programos procedūra arba nuo kurio priklauso procedūros (paprogramio) vykdymas.

Lange *Startup* nenurodyta labai svarbi savybė - *AllowByPassKey*, suteikianti galimybę klavišu *Shift* ignoruoti arba vykdyti *Startup* nustatytus parametrus.

2.6. Duomenų bazės savybių nustatymas

Duomenų bazės savybių langui išvesti komanda nurodoma duomenų bazės lange: *File* → *Database Properties* (arba iš parankinio komandų sąrašo). Duomenų bazės administravimo savybės sudaro keletą objektų *Document*, kuriuos valdo branduolys *Jet*. Yra penkios administravimo kategorijos, žr. kortelės: *General*, *Statistics*, *Summary*, *Contents*, *Custom*. Pirmąsias tris galima tik skaityti.



2.25 pav. Duomenų bazės *DB_EXPENSES* savybių kortelių/puslapių rodinis galimas matyti, kai *DB* pavadinimo eilutėje spustelimas pelės dešinys klavišas ir pasirinkama komanda *Database Properties*

Kortelėje *General* pateikiama ta pati informacija, kurią galite matyti *Windows Explorer* lange, kai ties duomenų bazės vardu spustelsite pelės dešinįjį klavišą ir nurodysite komandą *Properties*.

Statistics - matysite duomenų bazės parengimo datą ir laiką, paskutinio modifikavimo, naudojimo ir spausdinimo datą ir laiką.

Contents - pateikiamas duomenų bazės pagrindinių objektų vardų sąrašas.

Summary - pateikia blokus informaciniais tekstams įvesti. Įvedama informacija saugoma kaip savybės *SummaryInfo* reikšmė.

Custom - leidžia parengti specialios paskirties duomenų bazės savybes, pvz., nurodyti galimybę panaudoti duomenis.

2.17 UŽDUOTIS.

Reikia išvesti visais galimais būdais ir peržiūrėti duomenų bazės *DB_EXPENSES* savybių sąrašus:

1. Nurodyti komanda *File* → *Database Properties*.
2. Nurodyti komanda *Database Properties* iš parankinio komandų sąrašo (kai pagrindiniame DB lange pelės rodyklė ties DB vardu, spustelti dešinį klavišą).
3. *Windows Explorer* lange, ties duomenų bazės vardu spustelti pelės dešinįjį klavišą ir nurodyti komandą *Properties*.

2.7. Sistemos Access aplinkos nustatymas

Tinkamai nustatydami *Access* aplinkos parinktis *Options* paspartinsite veikimą.

Kai nurodysite komandą: *Tools* → *Options*, atvertame lange pastebėsite 8 kortelės: *View*, vaizdavimas; *General*, bendrosios parinktys; *Edit/Find*, redaguoti/rasti; *Keyboard*, klaviatūra; *Datasheet*, įrašų lentelė; *Forms/Reports*, formos/ataskaitos; *Advanced*, papildomos galimybės; *Tables/Queries*, lentelės/užklausos. (*Access XP* yra dar: *Pages*, *International* ir *Spelling*, o *Access 97* atskiros kortelės: *Module*, modulis/programa ir *Hyperlinks/HTML*, hipersaitai/HTML).

Parinktis, pasirinktis option - numatyta papildoma apdorojimo priemonė programos darbo pagrindiniam režimui modifikuoti.

Visos aplinkos parinktys (daugiau 60) interpretuojamos ypatingu būdu ir saugomos *Windows* centriniame informacijos banke, kurio vardas – *Registry*, registratūra. Pagrindiniai registratūros *Registry* dėmenys vadinami raktais. *Registry* turinį galite peržiūrėti duomenų banko lange; panaudodami teksto redaktorių *Registry Editor* (rinkmena *regedit.exe*): *Start* → *Run* → *regedit.exe* → *OK*. **DĖMESIO! Nieko nekeiskite!**

Redaktoriaus lange *Registry Editor* matysite raktų sąrašą. Raktuose (aplankuose) yra informacija apie kompiuterio aparatūrą, išorinius įrenginius, programų parametrus ir t.t. Raktai (knygos) turi papildomus raktus ir duomenis, t.y. reikšmes.

Raktas - *key* - duomenų elementas (laukas arba jų grupė), identifikuojantis įrašą (failą) ar nurodantis buvimo vietą.

Access aplinkos parinkčių reikšmės galima pamatyti atvertus papildomojo rakto langą *Settings* (*Access 97* adreso eilutėje 8.0, *Access 2000* - eilutėje 9.0):

HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Access\Settings

Access įterptųjų komandų reikšmės taip pat saugomos banke *Windows Registry*. Komandas galima pamatyti papildomojo rakto lange *CommandBars*:

HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Access\CommandBars

Dažniausiai *Access* aplinkos parinktys nustatomos dialogo lange *Options*, bet daugelį jų galima nurodyti *VBA* procedūrose.

2.8. Apsaugos parametrų nustatymas

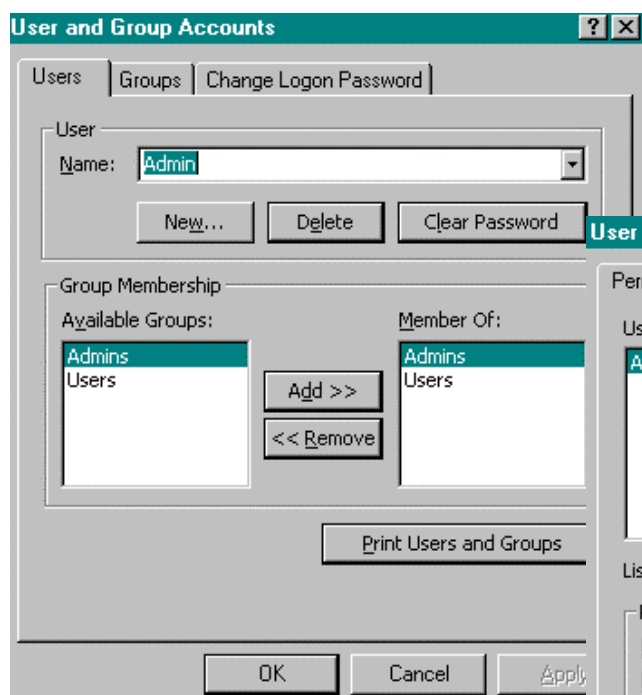
Access apsaugos priemonės valdo duomenų bazės branduolys *Jet*. Taikomi du apsaugos būdai: apsauga slaptažodžiu ir apsauga darbo grupės lygiu. *Access* projektų, kurių duomenys yra serveryje, apsauga organizuojama kitaip.

Apsauga - *security, protection* - priemonė kreiptis į duomenis ar kompiuterio bei jo dalies naudojimui riboti; juridinės, organizacinės, programinės ir techninės priemonės nesankcionuotoms kreiptis į aparatūrą, programas ir duomenis išvengti.

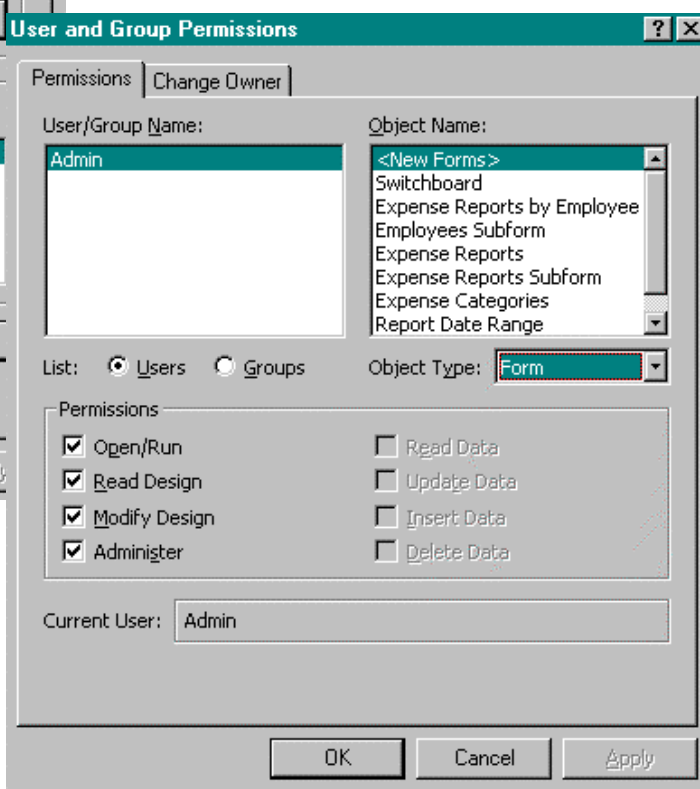
Darbo grupių apsaugos būdas numato dvi apsaugos dedamąsias: vartotojus ir įgaliojimus. Vartotojai atpažįstami vardais ir slaptažodžiais. Veiksmams su duomenų bazės duomenimis numatomi įgaliojimai arba leidimai, pavyzdžiui, *ReadData* arba *ModifyDesign*. Informacija apie įgaliojimus saugoma duomenų bazėje, o žinios apie vartotojus – darbo grupės informacijos rinkmenoje – *system.mdw*.

Darbo grupė – tai kartu dirbančių su duomenų bazės duomenimis darbuotojų grupė.

Informacija apie vartotojus, grupes ir slaptažodžius galima įvesti, saugoti ir, pagal reikalą, keisti panaudojant komandos *Tools* → *Security* tęsinius: *User and Group Accounts*, *User and Group Permissions*, *User – Level Security Wizard*. Galima nustatyti įgaliojimus/leidimus kiekvienam pagrindiniam duomenų bazės objektui ir parengti įgaliojimų/leidimų rinkmeną.



2.26 pav. Dialogo lange *User and Group Accounts* galima skelbti informaciją apie vartotojus, grupes ir slaptažodžius: paskirti arba keisti naujus vartotojus, sukurti vartotojų grupes ir suteikti jiems slaptažodžius.



Leidimą vartotojui dirbti su atskiru duomenų bazės objektu arba dirbti su keliais objektais galima nustatyti dialogo lange *User and Group Permissions*

2.18 UŽDUOTIS.

Tyrinėti ir nustatyti tinkamus darbui su DB objektais įgaliojimus, leidimus ir rengti informacinę rinkmeną taikant aukščiau minėtas parinktis: *User and Group Accounts*, *User and Group Permissions*, *User – Level Security*

Wizard bei išbandyti (po vieną) pasirenkamų nuostatų veikimą. Užsirašykite dialogo languose padarytus keitimus ir slaptažodžius.

Patarimas: kadangi teks dirbti su nepakankamai aiškintomis nuostatomis, be to, dar neturite reikiamos patirties, DB neprarasite ir išvengsite kitų nemalonumų, jeigu dirbsite su DB kopija. (Darbo aplanke PIRMADALIS-2sk ties *DB_EXPENSES* vardu spustelti pelės dešinį klavišą, *Copy*, dar kartą dešinį klavišą, *Paste*; matysite *Copy of DB_EXPENSES*; pakeiskite DB vardą – spustelti dešinį klavišą, *Rename* ir vietoj *Copy of* surinkti *Kopija*; dabar jau galima dirbti su duomenų bazės *DB_EXPENSES* kopija, kurios vardas – *Kopija DB_EXPENSES*. Sėkmės!)

2.9. Įgaliojimų saugojimas objektuose *Document* ir *Container*

Kiekvienas pagrindinis duomenų bazės objektas, kurių vardai rodomi kortelėse *Tables*, *Queries*, *Forms* ir t.t., turi atskirą kreipties į duomenis objektą, saugomą bendrame duomenų bazės kreipinių į duomenis objekte, kuris vadinamas *Document*.

Duomenų bazės taikomosios programos branduolys *Jet* panaudoja objektą *Document* nustatytiems įgaliojimams ir leidimams tikrinti, taip pat yra tikrinama objekto parengimo data ir tikrinamas parengėjas-vartotojas.

Be to, kiekviena pagrindinių duomenų bazės objektų kategorija, pavyzdžiui, formos arba makrokomandos turi kreipties į duomenis atitinkamą objektą - objektą *Container*. Duomenų bazės branduolys *Jet* panaudoja objektą *Container* tikrinant pagrindinių objektų vardus, taip pat įgaliojimus bei leidimus. Yra ir atskiras atvejis: du pagrindiniai duomenų bazės objektai - lentelės *Tables* ir užklausos *Queries* - turi vieną bendrą kreipties į duomenis objektą *Container*, kurio vardas *Tables*.

Nustatomi kreipties į duomenis įgaliojimai ir leidimai (*permissions*) yra ne kas kita, kaip kreipties į duomenis objektų *Document* ir *Container* savybės.

2.10. Objektų egzistavimo laiko trukmė

Kalbame apie objekto egzistavimo laikotarpį. Kai sukuriame pagrindinius duomenų bazės objektai: *Tables*, *Queries*, *Forms* ir t.t., jie išsaugomi duomenų bazės rinkmenoje, kurio vardo plėtinys *.mdb*. Tai pastovieji arba nuolatiniai duomenų bazės objektai. Jų egzistavimas prasideda parengimu ir baigiasi, kai pašalinami iš duomenų bazės rinkmenos. Ne visi objektai, kuriuos valdo duomenų bazės branduolys *Jet* yra nuolatiniai. Kai kurie kreipties į duomenis objektai, pradedant *DBEngine*, *Error*, *Workspace* ir *Recordset* yra laikini ir rinkmenoje *.mdb* neregistruojami. Kai kurie laikinieji objektai parengiami automatiškai, kai atveriamą duomenų bazės rinkmena, kiti - parengiami *VBA* procedūrose. Tokie objektai egzistuoja tik kompiuterio operatyviojoje atmintinėje.

Laikinis objektas gali egzistuoti visą duomenų bazės darbo laiką - nuo įkrovimo iki uždarymo arba trumpesnį laikotarpį. Pavyzdžiui, *VBA* procedūroje objektas *Recordset* egzistuoja tik procedūros parengimo ir vykdymo metu.

Vėliau susipažinsime, kaip parengti *VBA* taikomas programas objektams valdyti. Programavimas panaudojant branduolio *Jet* galimybes įgalina nuolat kurti kreipties į duomenis objektus tada, kai jie reikalingi, ir šalinti, kai nebereikalingi.

2.11. Objektų valdymo programos

Programų paskirtis - kompiuteriui teikti veiksmų su objektais komandų sekas. Yra dviejų rūšių programos: *įterptosios*, t. y. *Access* *vidinės* *programos*, ir *vartotojo parengtosios* *programos*. Įterptosios programos paleidžiamos nurodant *Access* komandas iš komandų sąrašų arba komandų mygtukais, esančiais mygtukų juostose. Kitos - įterptosios vidinės programos veikia automatiškai ir sudaro galimybę sistemai *Access* tinkamai funkcionuoti. Savos programos paleidžiamos specializuotų komandų sąrašų komandomis arba į objektų langus įterptaisiais komandų mygtukais.

Kitaip apibrėžiant komandų paskirtį, galima teigti, kad programos manipuliuoja objektais. Programos atveria ir uždaro formas, vykdo užklausas, spausdina ataskaitas, išsaugo įvairius pakeitimus duomenų bazės rinkmenoje, nustato paleisties parametrus ir t.t.

Programos pagal jų saugojimą objektuose gali būti vidiniais arba išoriniais *o b j e k t ų k o m p o n e n t a i s*:

Programos, kurios saugomos kaip vidiniai objektų komponentai, yra vadinamos *m e t o d a i s*. Į objektus galima kreiptis įterptaisiais metodais arba savo parengtais metodais.

Programos, kurios saugomos ne objektuose, bet atskirai nuo objektų, vadinamos makrokomandomis ir *VBA* procedūromis. Programos atlieka veiksmus su objektais.

Pastaba. Formos arba ataskaitos modulio procedūros, saugomos modulyje, nepriskiriamos nei vienai tik ką paminėtai programų kategorijai. Jeigu tai savos procedūros (*private*), tai jas galima vadinti formas arba ataskaitos metodais. Kitos bendrosios procedūros (*public*) - yra išoriniai komponentai, net jeigu jos saugomos formoje arba ataskaitoje. Apie tai smulkiau sužinosite vėliau.

2.12. Makroveiksmų komandų panaudojimas

Makrokomandų programose objektais manipuluojama makroveiksmų komandomis. *Access* sudėtyje yra apie 50 makroveiksmų komandų.

Makroveiksmų komandos optimizuojamos individualiai, siekiant didžiausio našumo. Vienos makroveiksmų komandos yra ekvivalentiškos komandų sąrašų komandoms (tokios, kaip *Close*). Kitos imituoja vartotojo vykdomas operacijas (pvz., *SelectObject*). Trečios siūlo galimybes, kurių nėra vartotojo sąsajų srityje (pvz., *Beep*).

Daugiausia makroveiksmų komandos panaudojamos duomenų bazės objektams ir dėmenims valdyti. Galima atverti, uždaryti, perkelti, įkelti, iškelti, šalinti objektus ir t.t. Vienas iš makroveiksmų - *SetValue* panaudojamas objektų savybėms keisti.

Sistema *Access* atskirų makroveiksmų komandų rengimas nėra numatytas.

Makroveiksmų komandų darinį įprasta vadinti makrokomanda. *M a k r o k o m a n d a* - tai vienisiai vykdoma makroveiksmų komandų seka. Makrokomandų instrukcijose atspindimas interaktyvus darbas. Interaktyvus operacijų žingsnių vykdymas ir žingsnių keitimas makroveiksmų komandomis yra geriausias būdas makrokomandomis nagrinėti. Dirbant interaktyviai pirmiausia nurodomas objektas, po to vykdomas veiksmas su juo, pvz., langas pažymimas bakstelint pele, o valdymas vykdomas tabuliacijos klavišu arba taip pat pele. Norint pažymėti įrašą reikia bakstelėti *s e l e k t o r i ų* - eilutės mygtuką arba nurodyti panaudojant komandą *GoTo* iš komandų sąrašo *Edit*.

Vadinasi, makroprogramų rengimas (makroprogramavimas) yra orientuotas į pasirinkimo operaciją - reikia nustatyti, koks yra nurodytas objektas. Tai galima atlikti dviem būdais:

Pasinaudoti makroveiksmu, nurodančiu objektą, pvz., *OpenForm*, *SelectForm* arba *GoToControl*.

Pažymėjus objektą pasinaudoti veiksniaus/aktyvaus objekto savybėmis, pvz., *ActiveForm* ir *ActiveControl*.

Makroprogramose metodai (t. y. programos, saugomos kaip vidiniai objektų komponentai) nenaudojami ir apsiribojama tik objektų savybėmis.

Veiksnumas arba **aktyvumas** - tai objekto galimybė reaguoti į pelės arba klaviatūros operacijas (tas pats fokusuotė). Visuomet fokusuojamas tik vienas valdomasis elementas. Objekto veiksmi/aktyvi būklė (arba fokusuotė) matoma vizualiai, pvz., į tekstinio valdomojo elemento lauką įkeltas įvesties žymiklis arba komandų mygtuko pavadinimas ant mygtuko apvestas punktyriniu stačiakampiu. Kai kurie valdomieji elementai, kaip antai, ženklai, įprastinės linijos, stačiakampiai, puslapių žymenys ir kt., negali tapti veiksniais (jų negalima fokusuoti). Veiksnūs (fokusuojamieji) valdomieji elementai turi savybes *Enabled* ir *Visible*.

Fokusuotė, fokusavimas – *focus, focussing* – valdomojo elemento aktyvinimas.

2.13. VBA procedūrų panaudojimas

VBA programavimo kalba manipuliavimą objektais galima reikšti dviem būdais:

- Pirmas būdas. Panašiai kaip makroprogramose - objektui veikti išskviečiama objektų išorinė programa. Išorinės programos analogiškos makroveiksmų komandoms, bet nelaikomos elementariomis veikimo programomis, kaip makroprogramose, o apibūdinamos kaip *Access VBA specialaus objekto metodu* su vardu *DoCmd*. Metodai *DoCmd* atitinka daugelį makroveiksmų todėl ir jų veikimas yra panašus.
- Antras būdas. Sudaroma galimybė manipuluoti betarpiškai objektu ir panaudojant jo vidines programas. Šis būdas vadinamas *objekto principu*. Objekto savybės nustatyti arba jas keisti galima priskyrimo operatoriumi. Be to, galima panaudoti objekto savus metodus. *Metodu* - tai *Access* įterptiniai ir iš anksto numatomi objekto veiksmai, saugomi kaip objekto vidiniai komponentai.

VBA objektas apibūdinamas savybėmis, nusakančiomis, koks yra objektas, ir metodais, nurodančiais ką jis turi vykdyti.

Objektams išrinkti/žymėti gali būti panaudojami metodai ir savybės:

- Metodai *OpenForm*, *SelectObject* arba *GoToControl*, objekto *DoCmd* ir valdomojo elemento arba formos metodus *SetFocus*.
- Objekto savybės *ActiveForm* arba *ActiveControl* (kai objektas jau pažymėtas/išrinktas).
- Savybė *Me*, jei reikia nurodyti formą arba ataskaitą, kurioje vykdoma procedūra.
- Be to, *Access* leidžia parengti VBA procedūras iš karto manipuluojančias objektais, be jų išankstinio išrinkimo. Tai *objekto orientavimas procedūros*. Jas parengti žymiai sudėtingiau, tačiau veikia žymiai sparčiau, nes nebereikia skirti laiko objektui išrinkti.

2.14. Sistemos Access įvykiai

Objekto būseną - tai jo savybių visumą duotuoju laiko momentu. Ji saugoma atmintinėje.

Būsena state - esama sistemos arba įrenginio (laisvas, užimtas, atjungtas), programos (eilėje, sulaikyta, aktyvi, užbaigta), failo (atidarytas, uždarytas, kataloguotas, pašalintas ir pan.) būklė.

Keičiant nors vieną objekto savybę keičiama objekto būseną. Kai kurie objekto būsenos pakeitimai suteikia galimybę nutraukti anksčiau nurodytą objekto apdorojimą. Tokie būsenos pakeitimai vadinami *įvykiais* (*events*). Kai atveriamą formą, keičiama būseną „neatverta“ į būseną „atverta“. Tokį duotosios formos būsenos pakeitimą nustato įvykis, vadinamas įvykiu *Open*.

Ne visi būsenos pakeitimai yra įvykiai. Pvz., formos fono spalvos pakeitimas nėra įvykis. *Access* nesuteikia galimybės vartotojui nustatyti savus įvykius.

Įvykis - *event* - tam tikro vyksmo atsiradimas arba pasibaigimas kompiuterinėje sistemoje; signalas apie tam tikro požymio būseną.

Į *Access 97* yra įvesti keturi nauji įvykiai: *Initialize*, *Terminate*, *ItemAdded* ir *ItemRemoved*.

Įvykius *Initialize* ir *Terminate* atpažįsta (*recognize*) objektas, vadinamas klasės moduliu. **K l a s i ū m o d u l i a i** - tai *VBA* išplėtimo priemonės, skirtos objekto paskirčiai nustatyti. Po to, kai objektas atpažintas ir įtrauktas į modulį, gali būti vykdoma procedūra objektui parengti atmintinėje.

Kai objektas įtraukiamas į modulį, klasės modulis atpažįsta įvykį *Initialize*. Kai naujas objektas nustoja egzistuoti, klasės modulis patvirtina įvykį *Terminate*.

ATPAŽINTIS, IDENTIFIKAVIMAS - *recognition, identification* - tapatybių nustatymas, pažinimas iš kokių nors ženklų; identifikatoriaus priskyrimas objektui.

Įvykius *ItemAdded* ir *ItemRemoved* atpažįsta (*recognize*) kitas objektas, kurio pavadinimas *References*, kreiptys, kai teikiama arba šalinama į/iš *Access* duomenų bazės nuoroda, į tipų biblioteka ar kitą taikomąją programą. **T i p ū b i b l i o t e k a** - tai rinkmena, esanti kitoje taikomoje programoje arba kitoje duomenų bazėje, kurioje pateikiamas objektų aprašas, taip pat savybės ir metodai. Su tokia taikomąja programa arba duomenų baze galima susisiekti panaudojant *VBA* procedūras.

Įvykiai *ItemAdded* ir *ItemRemoved* turi būti apdorojami tam tikru būdu. Makrokomandose nenaudojami.

Kiekvienam įvykiui ar tai formos, ar valdomojo elemento, ar ataskaitos įvykis - numatomos atitinkamos įvykių apdorojimo procedūros. Įvykių savybės vardijamos *Event* kortelės sąrašuose.

2.15. Access programavimo modelis

Access programavimas orientuojamas į įvykius. Kai analizavimo objektas atpažįsta įvykį, sukuriamas makrokomanda arba *VBA* procedūra. Įvykis paleidžia veikti programą, t.y. *Access* vykdo duotajam įvykiui suteiktą makrokomandą arba procedūrą, kuri vadinama **į v y k i o a p d o r o j i m o m a k r o k o m a n d a** arba **p r o c e d ū r a** (įvykio apdoroklis).

VAB būdingos dvi procedūrų rūšys: procedūros - funkcijos ir procedūros - paprogramės. **P r o c e d ū r a** - **p a p r o g r a m ė**, panaudojama įvykiui apdoroti, vadinama tiesiog įvykio apdorojimo procedūra, o **p r o c e d ū r a** - **f u n k c i j a** vadinama įvykio procedūra - funkcija.

Kai procedūrai suteikiama įvykio apdorojimo programa, kartu nustatoma įvykio gaudyklė. **Į v y k i o g a u d y k l ė** pakeičia standartinį *Access* veiksmą, į kurį vėl grįžtama po įvykio apdorojimo. Po kai kurių įvykių apdorojimo programa baigia darbą ir nebevykdo po įvykio esančių operacijų.

Access savo sudėtyje turi numatytus įvykius formoms ir ataskaitoms, jų dalims ir formų valdomiesiems elementams. Įvykių kreipties į duomenis, įvykių lentelėms ir užklausoms numatyta nėra. *Access 97* yra numatyti du įvykiai objektui *References*.

2.16. Įvykių kategorijos

Access aplinkoje yra numatyti net 44 įvykiai. Įvykius priimta klasifikuoti pagal jų pasireiškimo priežastis. Čia pateikiamos tik įvykių kategorijos: *Mouse events*, *Keyboard events*, *Window events*, *Focus events*. Kitos įvykių (*events*) kategorijos: *Filter*, *Print*, *Error*, *Timing*, *Reference*, *Class module events*.

Paskutiniosios dvi kategorijos būdingos tik *VBA*.

PRATIMAS.

Išsamią informaciją apie įvykius pateiks pagalbininkas-asistentas *Office Assistant*. Duomenų bazės lange bakstelėkite mygtukų juostoje mygtuką *Microsoft Access Help*. Asistento *Office Assistant* lango *What would you like to do?* įvesties laukelyje surinkite *Events* ir bakstelėkite *Search*. Po to nurodykite temą *Events: Making your database objects work together*. Lango pabaigoje bakstelėkite įterptą į tekstą priešpaskutinį mygtuką - *For a complete list of events, click*. Matysite apie įvykius išsamios informacijos langą *Events and Event Properties Reference*, įvykiai ir įvykių savybių žinynas. Peržiūrėkite informaciją apie kai kuriuos jau žinomus įvykius. Įsidėmėkite, kaip surasti aiškinimus. Galėsite paskaityti atidžiau, kai reikės.

2.19 UŽDUOTIS.

1. Iš kompaktinio disko įkrauti lentelių rinkmeną: *\MSO-2K\Lenteles\Chapter02.pdf*.
2. Peržiūrėti programos *Adobe Acrobat Reader* aplinkoje įvykių lenteles. Rasti ir kopijuoti lentelę *Access Events Grouped by Causes*, įvykiai sugrupuoti pagal kilmės priežastis, kopiją įterpti į *Word* langą.
3. Įrašyti dokumento-lentelės rinkmeną savo darbo aplanke.
4. Peržiūrėti lentelėje įvykių sąrašą.

Pastaba: lentelės rinkmenos *Chapter02.pdf*, kaip matyti formatas *.pdf*. Lentelė sėkmingai kopijuojama kaip grafinis objektas, kai prieš kopijavimą *Adobe Acrobat Reader* lange išskiriama rėmeliu (reikia bakstelėti mygtuką *Graphics Select Tool* ir apibrėžti lentelę).

2.17. Objektams būdingos įvykių savybės

Kadangi įvykiai apsprendžia objekto būsenos kitimą, įvykius galima klasifikuoti objektų atžvilgiu, pvz., įvykiai, atpažįstami duomenų formų, formų skyrių, formų valdomųjų elementų, ataskaitų bei ataskaitų skyrių.

Įvykių sekų nuoseklumas. Kada vartotojas, makrokomanda arba *VBA* procedūra vykdo kokius nors veiksmus, vienas objektas arba keletas objektų analizuoja ir atpažįsta (*recognize*) įvykių seką, pateikiamą tam tikru nuoseklumu/eiliškumu.

Bakstelėjimas pele. Spustelėjus pelės kairiąją klavišą ties valdomuoju elementu yra galima ir tikrinama trijų įvykių seka:

- *MouseDown* - pelės klavišo nuspaudimas.
- *MouseUp* - pelės klavišo atleidimas.
- *Click* - pelės kairiojo klavišo spustelėjimas/spragtelėjimas (nuspaudimas ir tuoj pat atleidimas – tai, ką vadiname „bakstelėti“).

Nepriklausomai nuo pelės klavišo, valdomasis elementas gali atpažinti (*recognize*) įvykius *MouseDown* ir *MouseUp*, tačiau įvykis *Click* fiksuojamas tik spragtelint kairiuoju klavišu.

Komandinio mygtuko bakstelėjimas pele. Jei valdomasis elementas gali tapti veiksmu/aktyviu, galimi papildomi įvykiai. Pavyzdžiui, kai pelės rodykle bakstelėjamas komandos mygtukas, pateikiama penkių papildomų įvykių nuosekli seka:

- *Enter* – iki komandinio mygtukas gaus valdymą nuo kito tos pačios formos elemento arba taps pirmuoju valdomuoju elementu, aktyvinamu atveriant formą.
- *GotFocus* – po komandinio mygtuko aktyvinimo.
- *MouseDown* – nuspaudžiant pelės klavišą.
- *MouseUp* – atleidžiant pelės klavišą.
- *Click* – kairiojo klavišo nuspaudimo ir atleidimo pasekmė.
- Panašiai įvykiai nuosekliai atpažįstami ir kitų valdomųjų elementų, tokių kaip tekstų blokai, parinkčių mygtukai, sąrašo blokai ir perjungikliai.

Teksto keitimas tekstiname ir kombinuotajame bloke. Klavišo spustelėjimu galima keisti tekstą. Ne visi klavišai generuoja simbolių kodus (pvz., *Tab* ir *Enter* jų negeneruoja). Žinome, kad *Windows* naudoja simbolių rinkinius *ANSI*, kuris nustato klavišų simbolių ir išvedamų į ekraną simbolių atitikimą. Jeigu klavišas keičia tekstą, teksto blokas atpažįsta keturių įvykių seką:

- *KeyDown* – spustelint bet kurį klavišą.
- *KeyPress* – spustelint klavišą, generuojanti simbolį iš rinkinio *ANSI*. (*ANSI* – *American National Standards Institute*- Amerikos nacionalinis standartų institutas).
- *Change* – atpažįstant sistemos *Access* keitimus, taikomus teksto blokui.
- *KeyUp* – atleidžiant klavišą.

Jeigu klavišas nekeičia teksto, įvykis *Change* nepasireiškia.

Jeigu bandysite keisti tekstą arba atnaujinti valdomąjį elementą nuspausdami *Enter* arba panašiai norėsite pereiti į kitą valdomąjį elementą arba įrašą, tolimesnė eiga visiškai priklausys nuo to, koks yra blokas – tekstinis arba kombinuotasis.

Kai klavišu keičiamas tekstas, tekstiname bloke, arba vykdomas koks nors veiksmas valdomajam elementui atnaujinti, kai nurodomas kitas formos elementas, tekstinis blokas fiksuoja du papildomus įvykius:

- *BeforeUpdate* – sistema *Access* atpažįsta tekstiname bloke pakeistą reikšmę betarpiškai prieš pakeitimą buferyje (tarpinėje atmintinėje).
- *AfterUpdate* – sistema *Access* atpažįsta tekstiname bloke pakeistą reikšmę betarpiškai po pakeitimo buferyje (tarpinėje atmintyje).

Kada tekstiname bloke arba kombinuotajame bloke keičiamas tekstas perduodant klavišo kodą, spustelint *Enter* ar pereinant prie kito elemento, sistema *Access* palygina duotąją reikšmę su reikšme kombinuotajame bloke. Jeigu reikšmės sąrašas nėra, kombinuotasis blokas atpažįsta įvykį *NotInList*. Po to viskas priklauso nuo savybės/charakteristikos *LimitToList*.

- Jei savybės *LimitToList* reikšmė *Yes*, *Access* nepatvirtina pakeitimo ir kombinuotasis blokas atpažįsta įvykį *Error*, nuosekli įvykių seka būtų tokia: *KeyDown*, *KeyPress*, *Change*, *KeyUp*, *NotInList*, *Error*.
- Jei savybės *LimitToList* reikšmė *No*, *Access* atnaujina kombinuotąjį bloką. Šiuo atveju galioja tokia įvykių seka: *KeyDown*, *KeyPress*, *Change*, *KeyUp*, *NotInList*, *BeforeUpdate*, *AfterUpdate*.

Be to, klavišų kodus galima perduoti programiškai, panaudojant makroveikimą *SendKeys* arba to paties vardo operatorių *VBA*. Veikiant programiškai pasireiškia tokia pat įvykių seka, kaip ir perduodant kodus klavišais.

1.18. Duomenų atnaujinimo procesas

Padarytiems keitimams aptikti/fiksuoti *Access* naudoja dviejų buferių/tarpinių atmintinių sistemą.

Kai valdomojo elemento pavadinimas, pvz., lauko vardas, išvedamas pirmą kartą, jis įrašomas į laikinąją atmintinę, į taip vadinamą - *į r a š o b u f e r į*. Jeigu į lauko vardą įvesite kitokius simbolius, *Access* įrašo juos į kitą laikinąją atmintinę – *v a l d o m o j o e l e m e n t o b u f e r į*. Po to buferių reikšmės palyginamos.

Jeigu yra pakeitimų, valdomasis elementas atpažįsta įvykį *BeforeUpdate*, *Access* atnaujiną valdomąjį elementą, perkopijuodama duomenis iš valdomojo elemento buferio į įrašo buferį ir po viso to, valdomasis elementas atpažįsta įvykį *AfterUpdate*. Jeigu pakeitimų nėra – elementas neatnaujinamas.

Išsaugant įrašą *Access*, palygina jo duomenis (įrašo buferio turinį) su lentelėje saugomais duomenimis. Esant pakeitimui, forma atpažįsta įvykį *BeforeUpdate*, *Access* atnaujiną įrašą, kopijuodama iš įrašo buferio pakeistus duomenis į lentelės laukus, po to forma atpažįsta įvykį *AfterUpdate*. Jei duomenų keitimo nebuvo, atnaujinimas nevykdomas.

Parinkčių *Options* reikšmių keitimas. Kai pele keičiamos parinktys bakstelint kurį nors parinkčių valdomąjį elementą, pvz., mygtuką ar jungiklį, vykdoma tokia įvykių seka:

- *BeforeUpdate* – jei *Access* nustato, kad yra parinkčių grupėje pakeitimų betarpiškai prieš atnaujinimą pakeistų duomenų įrašų buferyje.
- *AfterUpdate* – kai atpažįstamas keitimas parinkčių grupėje betarpiškai po atnaujinimo pakeistų duomenų įrašų buferyje.
- *Click* – tuoj pat po įvykio *AfterUpdate*.

Perėjimas nuo vieno valdomojo elemento prie kito. Tabuliacijos klavišu *Tab* pereinant/peršokant nuo vieno valdomojo elemento prie kito abu valdomieji elementai patiria tokių įvykių seką:

- Prieš paliekant pirmąjį valdomąjį elementą, dar iki jis taps neveiksniu, elementas atpažįsta įvykį *Exit*.
- Tapęs neveiksniu pirmasis elementas atpažįsta įvykį *LostFocus*.
- Antrasis valdomasis elementas, į kurį pereinate, dar prieš tai, kai jis taps veiksnium, atpažįsta įvykį *Enter*.
- Kai antrasis valdomasis elementas tampa veiksnium, jis atpažįsta įvykį *GotFocus*.

Jeigu prieš išeinant iš valdomojo elemento buvo pakeisti duomenys, pirmasis elementas prieš aukščiau fiksuotų įvykių seką, atpažįsta dar du įvykius:

- Prieš duomenų atnaujinimą įrašo buferyje pirmasis elementas įgauna įvykį *BeforeUpdate*.
- Po duomenų atnaujinimo įrašo buferyje pirmasis elementas atpažįsta įvykį *AfterUpdate*.

Formos atvertis. Formos atverties procese fiksuojama tokia įvykių seka:

- *Open* – atveriant formą betarpiškai prieš pirmojo įrašo išvedimą.
- *Load* – po įrašų pateikties ir vaizdavimo iš atmintinės.
- *Resize* – esant pirmajai formos pateikčiai.
- *Active* – kai forma tampa veiksnium (išskyrus tą atvejį, kai duotoji forma yra valdoma kitos formos, kurios savybės *PopUp* reikšmė yra nustatyta *Yes*).
- *GotFocus* – kai formai suteikiamas valdymas, bet visi valdomieji elementai blokuojami/uždrausti arba paslėpti.

- *Current* – prieš tampant dabartiniu, t.y. veiksmu pirmajam įrašui.

Kai formoje yra nors vienas valdomasis elementas, ji neatpažįsta įvykio *GotFocus*. Vietoj to pirmasis matomas teisėtas valdomasis elementas įgalinamas tokiai veiksmų sekai:

- *Enter* – prieš valdomajam elementui tampant veiksmu.
- *GotFocus* – kai valdomasis elementas tampa veiksmu.

Programose, kurios yra valdomos įvykiais, tenka remtis geriausiai tinkančiais įvykiais. Įvykių klasifikavimas ir jų savybės yra pateikiamos žinyne *Help*, žr., *Events and Event Properties Reference*.

Be anksčiau minėtų įvykių, klaidų įvykiai gali pasireikšti, kai aptinkamos sąsajos/interfeiso arba duomenų bazės branduolio *Jet* klaidos. Kitu atveju gali būti generuojami nustatyto laiko intervalo pabaigos įvykiai. Žr. *Help* įvykių grupė: *Error and Timing Events*.

2.19. Nustatytosios standartinės elgsenos atšauktis

Kai objektas atpažįsta įvykį, sistema *Access* vykdo nustatytąją standartinę, t. y. pagal nutylėjimą, elgseną (*behavior*).

Nustatytoji standartinė elgsena- *specified default behavior* - sistemos standartinis veikimo būdas, kai nėra vartotojo nurodymų.

Pavyzdžiui, kai komandos mygtukas atpažįsta įvykį *Click*, sistema *Access* tikrina ar yra įvykiui *OnClick* numatyta makrokomanda. Jeigu nėra, sistema laukia kito veiksmo. Kitu atveju sistema *Access* vykdo pagal nutylėjimą seriją operacijų prieš įvykio makrokomandą arba po įvykio makrokomandos. Pavyzdžiui, pakeičiant valdomojo elemento duomenis arba tabuliacijos žingsniu pereinant į kitą valdomąjį elementą, atpažįstamas įvykis *BeforeUpdate*. Tada pagal nutylėjimą *Access* veikia tokiu būdu:

1. Atnaujina valdomąjį elementą pagal buferio/tarpinės atminties įrašą.
2. Atnaujintas valdomasis elementas atpažįsta įvykį *AfterUpdate*.
3. Atnaujintas elementas fiksuoja įvykį *Exit*.
4. Modifikuotas elementas atpažįsta įvykį *LostFocus*.
5. Kitas valdomasis elementas atpažįsta įvykį *Enter*.
6. Kitas valdomasis elementas atpažįsta įvykį *GotFocus*.

Kai kuriais įvykių atvejais (tame tarpe *BeforeUpdate*) *Access* pirmiau vykdo priskirtą makrokomandą arba *VBA* procedūrą ir tik po to vykdo nustatytąją pagal nutylėjimą elgseną. Tokiose makrokomandose arba procedūrose galima numatyti žingsnius, atšaukiančius veiksmus, kuriuos sistema įprastai vykdo pagal nutylėjimą.

Kompaktinio disko, CD “Nastolnaja knyga po Access 97 Macro&VBA”, kuris platinamas kartu su knyga “Access 97. Rukovodstvo po makrojazyku i VBA. C. Novalis. – M.: LORI. 1998” kataloge Tables, rinkmenoje *Chapter2.pdf*, lentelėje *TABLE 2.16: The events for which you can cancel subsequent default behavior*, yra nurodyta 15 įvykių, kurių vyksmas, nurodytas pagal nutylėjimą, gali būti atšauktas makrokomandose arba procedūrose įterptomis komandomis, tai įvykiai: *ApplyFilter*, *BeforeDelConfirm*, *BeforeInsert*, *BeforeUpdate*, *DblClick*, *Delete*, *Filter*, *Format*, *KeyPress*, *MouseDown*, *NoData*, *Open*, *Print*, *Exit*, *Unload*.

(Be to, mok. leid. KD ir įvykių žinynas, žr.: *Help, Events and Event Properties Reference*.)

2.20 UŽDUOTIS.

1. Iš kompaktinio disko „ACCESS DB automatizavimas“ įkrauti lentelių Chapter2.pdf rinkmeną: *MSO-97\Lenteles\Chapter2.pdf*.
2. Peržiūrėti pirmas 7 lenteles, matysite rekomenduojamus objektų vardams fragmentus/požymius (žr.: 2.1. Skyr., vardinė pastraipa „Objektų vardai“).
3. Rasti, paaiškinti objektų požymius: frm, rpt, mcr, bas, lbl, txt, cbo, cmd; objekto tipą: ctl, ctls, col, app, db, err, idx, fid, prp ir parengti atskirą Word dokumento rinkmeną, įrašyti savo darbo aplanke. 4. Rasti lentelę The events for which you can cancel subsequent default behavior, įvykiai, kurių standartinę nuostatą galima atšaukti, ir parengti kaip atskirą Word dokumento rinkmeną, įrašyti savo darbo aplanke. 5. Peržiūrėti lentelės įvykių sąrašą.

Įvykių sekų demonstravimas. Yra speciali papildoma programa *EventLogger*, parengta knygos „*Access 97 Developer's Handbook*“ autorių P. Litwin, K. Getz ir M. Gilbert (SVBEX, 1997), kurią panaudojant galima demonstruoti praktikos pratimus ir stebėti įvykių sekas.

(Kompaktinis diskas, CD „*Nastolnaja knyga po Access 97 Macro&VBA*“, taip pat ir KD „*ACCESS DB automatizavimas*“ kuriame galima rasti minėtą paslaugų programą (ir daugiau informacijos apie programavimą makrokalba ir VBA) yra platinamas kartu su knyga: *Access 97. Rukovodstvo po makrojazyku i VBA. C. Novalis. – M.: LORI. 1998*. Be to, naujesnė knyga: *Access 2000. Rukovodstvo po VBA. Novalis. – M.: LORI. 2001* rekomenduoja atsisiusti papildomas programas: www.lory-press.ru)

Programą *EventLogger* galima panaudoti autonomiškai arba įterpti į savo duomenų bazę. Knygos „*Access 97 Developer's Handbook*“ septintame skyriuje aiškinama, kaip padedant *EventLogger* peržiūrėti įvykius formose ir aiškinamos programos VBA procedūros.

PRATIMAS.

Autonomiškai panaudodami programą *EventLogger* peržiūrėkite demonstruojamus įvykių sekų sąrašus:

1. Kopijuokite iš kompaktinio disko (kompaktinis diskas *Web\Event Logger\EventLogger.mdb*) atsisiusta Interneto tinklu duomenų bazės aplanką *Event Logger* su rinkmena/failu *EventLogger.mdb* į savo kompiuterio kietojo disko aplanką, pavyzdžiui, *C:\My Documents\Macro&VBA\PIRMADALIS-2sk* (be abejo galima kopijuoti ir į kitą savo darbo aplanką). Atverkite duomenų bazės rinkmeną/failą *EventLogger.mdb*.

Prieš įkraunant pamatysite įspėjimą: The database 'EventLogger' is read – only. You won't be able to save changes mode to date or object definitions in this database (duomenų bazė tik skaitymui; pakeitimų išsaugoti negalėsite). Bakstelėkite OK. Lange Microsoft Access pele perstumkite vertikaliosios slankjuostės šliaužtę į apačią (jeigu reikia), pamatysite duomenų bazės lango pavadinimo juostą *EventLogger: Database*. Išskleiskite duomenų bazės langą – mygtukas *Maximize*. Su programa Access sutapdinto duomenų bazės lango Microsoft Access – [EventLogger:Database] kortelėje Forms pažymėkite formą *frmLog* ir bakstelėkite mygtuką *Open*. Formoje *frmLog* registruojami įvykiai pagal įvykių tipus. Formos lango Microsoft Access – [EventLog] apačioje matomi nustatomieji jungikliai *Filter Events*, atitinkantys įvykių tipus:

- Mouse Movement, pelės postūmis. Įvykis *Mouse Move*
- Mouse Click, spragtelėjimas pelės klavišu. Įvykiai: *Click*, *DelClick*, *MouseDown*, *MouseUp*.
- Data, duomenys. Įvykiai: *AfterDelConfirm*, *AfterInsert*, *AfterUpdate*, *BeforeDelConfirm*, *BeforeInsert*, *BeforeUpdate*, *Change*, *Current*, *Delete*.
- Keystroke, klavišo spustelėjimas. Įvykiai: *KeyDown*, *KeyPress*, *KeyUp*.

- Focus, aktyvinimas/neaktyvinimas. Įvykiai: Activate, Deactivate, Enter, Exit, GotFocus, LostFocus.
- Action, veiksmas. Įvykiai: Close, Error, Load, Open, Resize.
- Timer, laikmatis. Įvykis Timer.

2. Lange Microsoft Access – [Event Log] įjunkite jungiklius Mouse Click, Data, Focus ir Action. Bakstelėkite darbo lango srities viršuje, eilutėje Events. Matysite dabar dar tuščią įvykių eilutę, pažymėtą punktyrine linija.

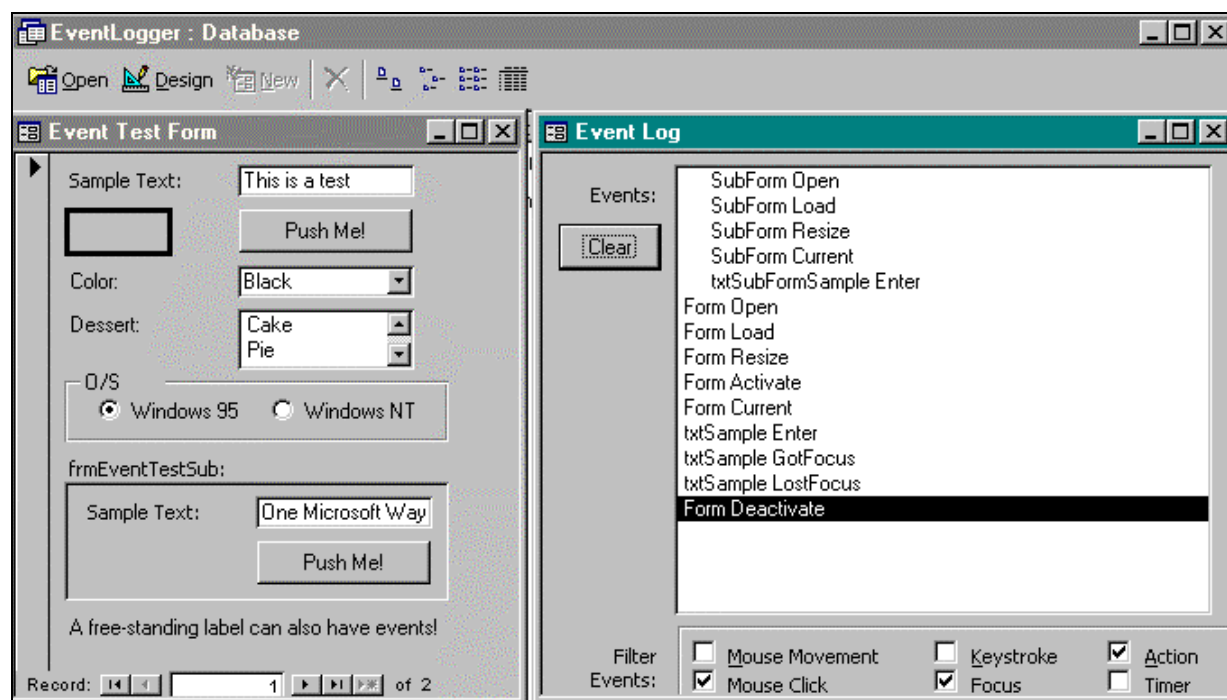
Kitą kartą, kai norėsite pašalinti vaizduojamą įvykių sąrašą, bakstelėsite mygtuką Clear. Dabar bakstelėkite Clear, ir vėl bakstelėkite eilutėje Events.

3. Vėl atverkite duomenų bazės EventLogger langą. Lange Microsoft Access - [Event Log] išskleiskite komandų sąrašą Window ir nurodykite EventLogger: Database. Atverkite formą frmEventTest. Tiesiog pažymėkite vardą ir Open.

Kokių įvykių seką generavo programa atverdama formą Event Test Form pamatysite, kai dabar atversite formą Event Log.

Iš komandų sąrašo Window nurodykite Event Log. Peržiūrėkite nuo pradžios įvykių sąrašą. Matome, kad pirmiausia atveriamą papildomoji forma, įkeliami jos įrašai, keičiami matmenys, atpažįstama dabartinė padėtis ir atveriamas tekstinis blokas (žr. pirmuosius penkis įvykius, prasidedančius priešdėliu Sub - tai: SubForm Open, SubForm Load ir t.t.).

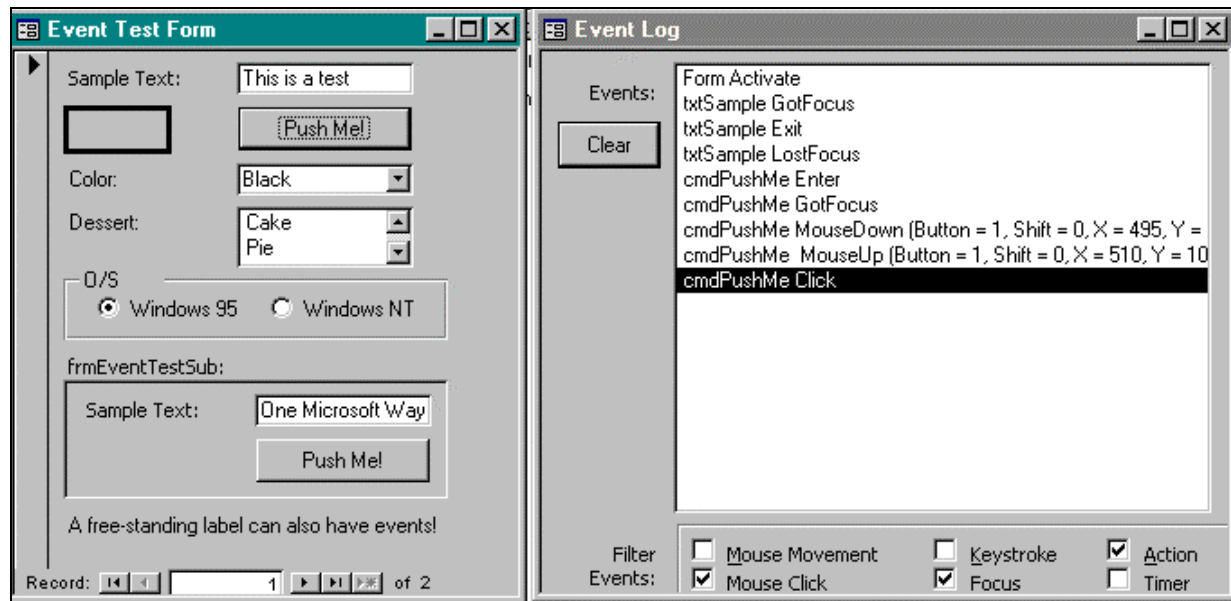
Po papildomosios formos pagrindinė forma atpažįsta savuosius atverties įvykius: Form Open, Form Load ir t. t. Pagaliau pirmasis pagrindinės formos aktyvus valdomasis elementas - tekstinis blokas txtSample įgauna įvykį GotFocus (txtSample Got Focus).



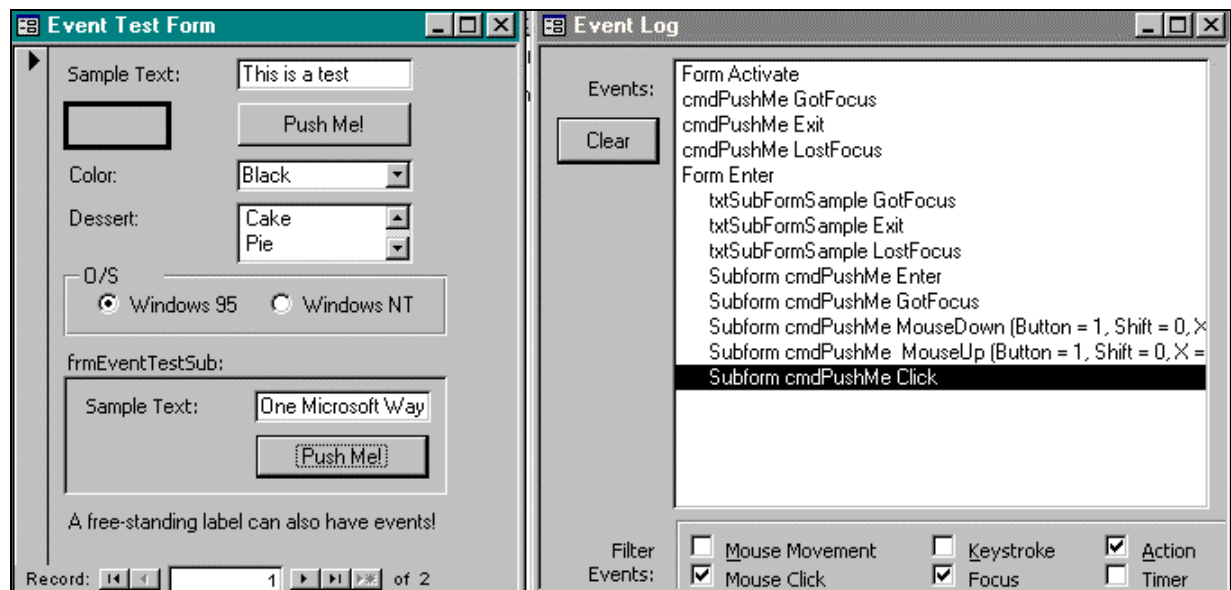
2.27 pav. Formos *Event Test Form* atvėrimas sužadina nuoseklių įvykių srautą, kuris matomas lange *Event Log*

4. Savybių sąrašo lange *Microsoft Access* - [Event Log] bakstelėkite mygtuką *Clear*. Vėl įkelkite duomenų formą *frmEventTest*. Nurodykite *Window* → *Event Test Form* ir pagrindinėje formoje bakstelėkite mygtuką *Push Me!* Atverkite formą *Event Log*. Nurodykite *Window* → *Event Log*. Matysite, kad forma *Event Test Form* atpažįsta įvykį *Activate* ir aktyvus tampa pagrindinės formos tekstinis blokas *txtSample*.

Tekstinis blokas atpažįsta įvykį *Exit*, po kurio seka įvykis *LostFocus*. Duotasis komandos mygtukas įgauna žemiau *cmdPush Enter* vardiniamą komandinių įvykių seką.



2.28 pav. Kai langas *Event Log* nuvalomas (mygtukas *Clear*), formos *Event Test Form* pagrindinėje dalyje bakstelint mygtuką *Push Me!* sužadinama nuosekli įvykių seka dėl pirmojo formos valdomojo elemento, kuris fokusuojamas pirmas, o po to pats mygtukas



2.29 pav. Kai vėl nuvalomas langas *Event Log*, bakstelint formos *Event Test Form* pagalbinėje dalyje mygtuką *Push Me!* sužadinama nuosekli įvykių seka formos pagrindinės dalies valdomajam elementui, kuris buvo fokusuojamas paskutinis, ir tik po to - formos pagalbinės dalies ir jos valdomajam elementui - mygtukui

5. Vėl bakstelėkite *Clear* ir formos *Event Test Form* lange bakstelėkite pagalbinės formos mygtuką *Push Me!* Pagrindinė forma įgauna įvykį *Activate*, po to aktyvus tampa valdomasis elementas, kuris gauna valdymą paskutinis - komandos mygtukas *Push Me!*

Pagrindinės formos mygtukas *Push Me!* atpažįsta įvykių seką, po to pagrindinė forma atveria redagavimui ir gauna savo valdomųjų įvykių įvykius. Pagalbinės formos valdomieji elementai įgauna įvykį *Exit* ir komandinis mygtukas *Push Me!* sau skirtą įvykių seką.

6. Patyrinėkite įvykius, kuriuos atpažįsta kiti formos *Event Test Form* valdomieji elementai. Pavyzdžiui, nustatykite *Windows NT* ir stebėkite mygtukams ir parinkčių grupėms būdingus

įvykius. Kombinuotajame bloke nurodykite kitą elementą. Redaguokite tekstą pagrindinės ir pagalbinės formos teksto sąrašuose ir t. t.

2.21 UŽDUOTIS.

1. Iš kompaktinio disko *ACCESS DB automatizavimas* aplanko *MSO-97\DBases* reikia kopijuoti aplanką *Event Loger* kartu su jame esama duomenų bazės (DB) rinkmena *EventLogger.mdb* ir įterpti į savo darbo aplanką kompiuteryje, pvz.: *C:\My Documents\Macro&VBA*.
2. DB *EventLogger* yra sukurta aplinkoje *Access 97*, todėl ją reikia pertvarkyti/konvertuoti į dabartinę, tarkim, *Access 2000*:
 - reikia panaikinti požymį *Read-only*, tik skaityti: kai pelės žymiklis ties vardu, spustelti dešinį klavišą, nurodyti *Properties*; skirsnyje *Attributes* išjungti jungiklį *Read-only*; bakstelti *Apply*;
 - paleisti veikti *Access*, uždaryti pagalbinį langą ir pagrindiniame lange nurodyti: *Tools → Database Utilities → Convert Database → To Current Access Database Version*;
 - lange *Database to Convert From*, rasti konvertuojamą DB, pažymėti ir bakstelti *Convert*;
 - lango *Convert Database Into* lauke *File name* surinkti naują vardą *Konvert-Logger.mdb* ir bakstelti *Save*;
 - konvertavimo procesui pasibaigus DB jau galima įkrauti: *File → Open*, pažymėti *Konvert-Logger.mdb* ir bakstelti *Open*;
 - jeigu DB lange nepamatysite, traukite žemyn lango dešinėje slankjuostės šliaužtę; apačioje kairėje pastebėsite minimizuotos DB lango ženklą, išskleiskite langą.
3. Atlikite pratimą su konvertuota DB *EventLogger* panašiai, kaip atlikote anksčiau. Dar kartą atidžiai patyrinėkite, vizualiai stebėkite, perskaitykite ir paaiškinkite įvykių sekas (2.27 – 2.29 pav.).

2.20. Santrauka

- Vardų sudarymo *vengriskasis stilius* leidžia iš vardo suvokti objekto tipą ir paskirtį.
- Savybės, parenkamos iš sąrašų, charakterizuoja objektą. Tai ypač svarbu projektavimo požiūriu. Kreipties objektai į duomenų bazės branduolio *Jet* duomenis savybių sąrašų neturi. Jų, taip pat ir objektų savybes galima nustatyti tik makrokomandose ir procedūrose. Kai kurias vykdymo etapų savybes galima tik skaityti.
- Dažnai tos pačios savybės gali būti taikomos makrokomandose ir *VBA* procedūrose, tačiau yra gana daug savybių, kurios taikomos tik procedūrose.
- Duomenų bazė taip pat žymima išskirtinėmis savybėmis. Galima nustatyti paleidimo parametrus, bendrąją informaciją ir duomenų apsaugos savybes.
- Sistemai *Access* galima nustatyti ir keisti aplinkos parametrus. Sistemos parametrai saugomi atskiroje informacijos rinkmenoje.
- Makroprogramavimas ryškiausiai skiriasi nuo programavimo *VBA* tuo, kad tik *VBA* sudaro galimybę kreiptis į metodus. *Metodas - tai įterptinė programa*, saugoma kartu su objektu, ir ją galima vykdyti atsižvelgiant į poreikius.
- Objekto būseną duotuoju laiko momentu nusako nustatytų savybių reikšmių visuma. Vartotojas, procedūros ir makrokomandos taip pat ir kompiuteris gali keisti objekto būseną. *Būsenos keitimai, vadinami įvykiais*, atveria

galimybę taikyti įvykiais orientuojamą programavimą. Įvykius atpažįsta tik formų valdomieji elementai, formų skyriai/dalys, ataskaitos ir ataskaitų skyriai.

- Dažniausiai vartotojo vienas veiksmas sukelia ištiesą seką įvykių. Labai svarbu įgyti reikiamą patirtį tinkamiausiems įvykiams pasirinkti.
- Makrokomandose ir *VBA* procedūrose panaudojami tokie pat įvykiai. Išimtis įvykiai *ItemAdded* ir *ItemRemoved*, kurie taikomi tik *VBA* procedūrose.

3. Sistemos Access objektinis modelis

- Objektų sąsaja
- Prietaikos *Application* ir duomenų bazės branduolio *Jet* objektų hierarchija
- Sistemos *Access* taikomosios programos objektinis modelis
- Nuorodos į objektus ir savybės pagal vardus
- Išraiškų komponuoklio *Expression Builder* panaudojimas nuorodoms rengti

Duomenų bazės automatizavimo operacijų komandos yra vykdomos objektams atpažįstant įvykius. Dėl to reikia žinoti kokiems objektams galima rašyti komandas ir kokius įvykius gali atpažinti objektas. Be to, reikia mokėti sudaryti makrokomandas ir procedūras.

Antrajame skyriuje buvo nagrinėjami objektai, kuriuos galima panaudoti arba pakeisti vykdant makrokomandas arba *VBA* procedūras.

Šiame skyriuje bus nagrinėjami objektų savitumai. Sužinosite, kurie sistemos *Access* objektai yra prieinami, kuriuos galima pakeisti, kaip jie sąveikauja, ir kaip taikomosiose programose reikia identifikuoti objektus.

3.1. Objektų sąsaja ir sąveika

Jau gerai žinote duomenų bazės lango dialoginės/interaktyviosios veiksėnos pagrindinius objektus: lenteles, užklausas, formas ir ataskaitas. Dabar objektų sąrašą galima papildyti lentelių ir užklausių laukais bei formų ir ataskaitų valdomaisiais elementais.

Taikant dialoginį interaktyvų darbą vartotojui visiškai nerūpi, kaip sąveikauja duomenų bazės objektai. Reikiamą objektų sąsają ir sąveiką užtikrina *Access* sistemos interfeisas/sąsaja.

Dialoginis, interaktyvusis – *conversational; interactive* – dialoginis/interaktyvusis darbo režimas; dvikryptė vartotojo ir sistemos sąveika.

Interfeisas – *interface* – bendr.: – duomenų rinkimo ir apdorojimo sistemos komponentų sąveikavimo sutarčių ir taisyklių sistema, kompiuterio techninės ir programinės įrangos priemonių tikslinga sietis ir sąveika.

Sąveika – *interaction; communication* – duomenų ir valdymo signalų mainai tarp sistemų, operacinės ir programinės įrangos; bendr.: - valdymo ir duomenų mainų procesas.

Norint programuoti, o programos, kaip žinia, manipuliuoja objektais, todėl reikia gerai žinoti objektų tarpusavio ryšius ir programose tinkamai panaudoti objektų savybes bei nuorodas.

Įvertinant objektų sąveiką bei akivaizdžius santykius gali būti **vieniniai objektai** panašūs/analogiški struktūros požiūriu (kada savo sudėtyje neturi kitų objektų) ir **sudėtiniai objektai**, kurių sudėtyje yra kiti objektai.

3.1.1. Analogiškųjų objektų grupės

Sakėme, kad analogiškieji/panašieji objektai skiriasi nuo kitų paprastesne struktūra, nes neturi savo sudėtyje kitų objektų. Tokius objektus su analogiškais savybėmis ir panašia elgsena įprasta jungti į atskiras grupes. Pavyzdžiui, galima jungti į grupes duomenų bazės formas, komandinius mygtukus arba teksto blokus. Panašių objektų grupę/sąranką vadinkime žodžiui „visuma“ analogišku žodžiu – visybė (visuma, visetas, visybė).

Visybė – *collection, assembly* – visa drauge, visuma. Daugelis *Access* objektų sudaro atskiras objektų visybes (*collection objects*). Pavyzdžiui duomenų bazė turi vieną visų lentelių visybę.

Kiekviena lentelė turi savo visų laukų visybę. Duomenų bazė turi vieną atvertų formų visybę, o kiekviena forma – valdomųjų elementų visybę.

Kiekvieno *Access* objekto vardas pradedamas didžiąja raide: *Form*, *Report*, *Control*, *Field*. Objekto visybės vardo pabaiga papildoma raide „s“ Konkretaus tipo valdomųjų elementų visybę *Controls*, turi būdingus formai valdomuosius elementus.

Pati visybė *Access* sistemos yra interpretuojama taip pat kaip objektas. Visybė *Controls* – tai objektas, turintis konkrečios formos/ataskaitos objektus *Control*.

Visybė *Forms* turi tik atvertas formas. Kiekviena atverta forma turi savo visybę *Controls*, kurioje yra būdingi formai valdomieji elementai ir gali būti dar kitos formos.

Visybė padeda atskirti atvirąją formą nuo uždarytosios formos. Atverta forma – tai objektas *Form*, o atvertų formų rinkinys sudaro visybę *Forms*. Uždaryta forma tai visai ne objektas *Form* ir nepriklauso visybei *Forms*. Uždaryta forma – tai tik uždaryta forma ir tiek.

Objektai, nepriklausantys visybėms yra vadinami išskirtiniais/singuliariniais.

Išskirtinumas/singuliariškumas – *singularity* – ypatingumas [lot. *singularis* – savotiškas, ypatingas]. Pavyzdžiui, yra objektas *Application*, pateikiantis sistemos *Access* taikomąją programą, ir objektas *DBEngine*, pateikiantis duomenų bazės (*DB*) branduolį *Jet*. Tai išskirtiniai/singuliariniai objektai, todėl kad *Access* sudėtyje yra tik vienas objektas *Application* ir vienas objektas *DBEngine*.

Visybės objektas taip pat pasižymi išskirtinumu/singuliariškumu. Pavyzdžiui, taikomojoje programoje yra tik vienas objektas - visybė *Forms* ir tos visybės kiekvienas objektas *Form* turi tik vieną būdingą visybę *Controls*.

Norint taikomojoje programoje nurodyti objektą labai svarbu žinoti ar tas objektas yra išskirtinis ar visybė.

Objektų grupės klasė ir egzemplioriai. Objektų grupė su analogiškais savybėmis ir elgsena yra apibūdinama objektų klase ir objektų egzemplioriais.

Klasė – tai objektų grupei būdingų objektų šablonas. Pavyzdžiui, formų klasė arba tekstinių sąrašų klasė. Lentelės struktūros pateikti lange *Design* galima priskirti lentelių klasei. Nustatant lentelių laukus ir jų savybes rengiamas lentelių klasės egzempliorius. Tekstinių sąrašų rengimo priemonė *Text Box* – tekstinių sąrašų klasė. Sudarant tekstinį sąrašą parengiamas tos klasės egzempliorius.

Klasė – *class* – bendrų požymių turinčių objektų klasifikacinė grupė.

EGZEMPLIORIUS – *copy, exemplor, specimen* – vienas iš daugelio tokių pat daiktų; vienetas.

3.1.2. Sudėtiniai objektai

Čia kalbame apie objektus, kurie savo sudėtyje turi kitus objektus. Tokių objektų tarpusavio priklausomybę, sąryšį nusako taip vadinamas įterpties sąryšis.

Sąryšis – *relation, interrelation, relationship* – ryšio tarp objektų išraiška, rodanti tų objektų bendrumo bruožus.

Įterpties ryšys – *inclusion relation* – tai ryšio tarp vyriausiojo objekto ir antrinių objektų išraiška.

Vyriausias objektas – *parent object* – vyresnumo ryšiais siejamų objektų, grandinės pirmasis/pagrindinis objektas.

Antrinis objektas – *child object* – vyresnumo ryšiais siejamų objektų grandinės priklausomasis (žemesnio lygmens) objektas. Objektas yra vyriausias jam priklausomiems objektams, o

priklausomieji objektai vadinami antriniais. Pavyzdžiui, forma yra vyriausias objektas formos valdomiesiems elementams (objektams), o valdomasis elementas – formos antrinis objektas.

Į objektų priklausomumo lygmenis galima žvelgti, kaip į hierarchijos lygmenis. Pažvelkime į lentelių įterpties sąryšio hierarchiją.

Sistemos *Access* objektas lentelė, vaizduojama konstravimo lange *Design*, vadinama objektu *TableDef*. Tai vienas iš duomenų objektų, valdomų branduolio *Jet*.

Lentelių visybės – *TableDefs* schema.

Schema vaizduoja lentelių hierarchiją. Kiekvienas stačiakampis – visybė, skliausteliuose vardinami visybių nariai. Kiekvieną visybę galima išplėsti ir parodyti dedamuosius objektus. Pavyzdžiui, duomenų bazės *Expenses* lentelių visybės *TableDefs* sudėtis: *Employees*, *Expense*, *Categories*, *Expense Details*, *Expense Reports*, *Switchboard*. Lentelės *Expense Categories* laukų *Fields* visybė: *EmployeeID*, *EmployeeName* ir t.t. Toliau kiekvieno lauko savybių *Properties* visybė ir kt.

Rašant programas tenka manipuluoti objektų savybėmis ir elgsena bei kreipiantis į objektus atsižvelgti į jų hierarchiją, todėl labai svarbu žinoti visų objektų įterpties ryšius.

3.2. Sistemos Access architektūra

Architektūra – *architecture* – sudėtingos sandaros elementų (loginių, fizinių, programinių ir pan.) sąryšio samprata.

Sistema *Access* sudaro du svarbiausieji komponentai: *Access Application* ir branduolys *Jet*. Darbui su *Access* projektais, kai panaudojama sistema *Microsoft SQL Server* reikia įdiegti duomenų bazės branduolį *Microsoft Database Engine MSDE*.

3.2.1. Sistemos Access programa Application

Sistemos *Access* taikomojoje programoje/sistemos pagrindinėje prietaikoje *Application* yra visos rinkmenos vartotojo sąsajos valdyti, makrokomandoms ir *VBA* procedūroms vykdyti. Programai priklauso komandų sąrašai, mygtukų juostos ir langai pagrindiniams duomenų bazės (DB) objektams rengti ir atvaizduoti. Interaktyviuoju duomenų bazės rengimo būdu parengti pagrindiniai objektai įterpiami į taikomąją programą. Nors *Application* sąsajos yra panaudojamos rengiant visus septinis duomenų bazės pagrindinius objektus, tačiau tik formos, ataskaitos, duomenų puslapiai (*Pages*) ir moduliai traktuojami kaip *Application* objektai.

Taikomosios programos *Application* sudėtyje yra duomenų bazės bibliotekos, įterpiniai, papildymai ir informacijos rinkmena.

DB biblioteka – tai duomenų bazės objektų ir procedūrų rinkinys. Bibliotekos rinkmenų plėtiniai yra *.mda* ir *.mde*. Yra penkios standartinės bibliotekos (pagal *Access 97*): *utility.mda*, *wzcnf80.mda*, *wzlib80.mde*, *wzmain80.mde* ir *wztool80.mde*. Biblioteka *utility.mda* atveriamą automatiškai, kai paleidžiama veikti sistema *Access*. Bet kurią kitą biblioteką galima atverti kaip įprastą duomenų bazę. Galima suformuoti savas bibliotekas.

Įterptiniai papildymai - *Add-ins* – priemonė konkrečioms užduotims vykdyti. Yra trys standartiniai įterptiniai papildymai (galima parengti ir savus).

- Pagalbinė programa – vedlys *Wizard*. Tai serija dialogo langų panaudojamų objektams rengti. Jos sudėtyje *Table*, *Query*, *Form* ir *Report Wizard*.
- Komponuoklis *Builder* taikomas elementams parengti. Galima konstruoti išraiškas arba nustatyti savybes.

Įterptinis komandų sąrašas, išvedamas *Tools* → *Add-ins*, pavyzdžiui, sužadinamos miniprogramos: *Switchboard Manager*, *Database Splitter*, *Menu Builder* ir *Linked Table Manager*.

Informacijos rinkmenoje *system.mdw* saugomos žinios apie vartotojus, slaptažodžius ir kt. Apsaugos informaciją galima nurodyti taip pat naudojant komandą *Tools* → *Security*.

3.2.2 Duomenų bazės branduoliai *Jet* ir *MSDE*

Duomenų bazės (DB) branduolio *Jet* sudėtyje yra rinkmenos duomenims valdyti, kreipties į DB objektų - lentelių ir užklausų duomenis informacijai teikti bei tikrinti ir programos *Application* objektams saugoti. Yra šešios įterptinės programos, kuriomis vykdomos duomenų bazės valdymo svarbiausios funkcijos:

- Duomenų nustatymas ir jų vientisumo užtikrinimas.
- Duomenų apsauga. Naudojamas indeksinis nuosekliosios kreipties metodas (*ISAM-Indexed Sequential Access Method*).
- Duomenų išrinkimas. Galimi du duomenų išrinkimo būdai: panaudojant užklausų struktūrinę kalbą (*SQL – Structured Query Language*) ir *VBA* procedūras.
- Manipuliavimas duomenimis. Panaudojamos užklausos *SQL* su veiksmų sekomis arba *VBA* procedūros.
- Apsauga. Galima visos duomenų bazės apsauga slaptažodžiu arba darbo grupių apsauga, kai leidžiama pasinaudoti tik kai kuriais objektais.
- Bendras duomenų naudojimas. Leidžiama dirbti vienu metu ir modifikuoti duomenis keliems vartotojams.

Jet leidžia kreiptis į visus objektus savarankiškai, t.y. nepriklausomai nuo duomenų bazės valdymo programų, panaudojant tiesioginės kreipties į objektus kalbą (*DAO – Data Access Objects*). Yra galimybė nustatyti betarpišką ryšį *ODBCDirect* su duomenų šaltiniu per *ODBC*. (*ODBC – Open Data Base Connectivity* – duomenų bazių standartinių ryšių sistema. Leidžia nusistatyti ryšį tarp *Access* sistemos ir kitų duomenų bazių sistemų, jeigu jose vartojama struktūrinė užklausų kalba *SQL*).

Duomenų bazės branduolys *MSDE*. *Access* projektuose (kai sukurama prietaika klientas-serveris, dirbanti, pavyzdžiui, su duomenų baze *Microsoft SQL Server*) panaudojamas naujas DB branduolys *MSDE*. Komponentų *OLE DB* architektūra (*OLE – Object Linking and Embedding* – objektų sietis ir įterptis) sudaro galimybę kreiptis į serverio duomenų bazę. Tai lyg vertėjas tarp objekto *Application* ir serverio.

Projektuose panaudojami 10 tipų objektai: lentelės, pateiktys, išsaugomos procedūros, diagramos, formos, ataskaitos, puslapiai, duomenų puslapiai, makrokomandos ir moduliai. Tačiau projektai savyje neturi nei duomenų objektų, nei duomenų valdymo objektų, neturi nei lentelių, nei pateikčių, nei diagramų, nei saugomų procedūrų. Tie visi DB objektai yra saugomi serverio DB, bet turi įtakos kitiems projekto objektams: formoms, ataskaitoms, duomenų puslapiams, makrokomandoms ir moduliams.

Lentelių saugojimo požiūriu pagrindinis skirtumas tarp duomenų bazių ir projektų yra duomenų lentelių saugojimo vieta. Projektų lentelės saugomos serveryje, duomenų bazių – *Access* aplinkoje. Kai nustatomas saitas su serveriu, vartotojas gali įprastu būdu operuoti duomenimis.

MSDE vykdo tokias pačias funkcijas kaip *Jet* (duomenų saugojimas ir apsauga, išrinkimas ir valdymas). Tam panaudojamas serverio funkcijų rinkinys, pvz.: *Microsoft SQL Server Design Tools*. Funkcijų rinkinys *Design Tools* yra įdiegtas į *Microsoft Access*. Funkcijos iškviečiamos sukuriant *Access* projekto objektus.

Naujas duomenų bazės branduolys *MSDE* įdiegiamas atskirai: *Office* įdiegimo kompaktinio disko aplanke `\Sql\86\Setup` du kartus bakstelint rinkmeną *Setupsql.exe*. (*XP – MSDE2000*).

3.2.3. Objektų hierarchija

Objektais manipuluoti galima makrokomandomis ir *VBA* procedūromis. *Access* ir *Jet* manipuliavimo operacijoms tinkami objektai vadinami įterptiniais objektais. Procedūromis galima parengti ir savus objektus.

Įterptiniai objektai grupuojami į visybes ir sudaro atskiras hierarchines priklausomybes. Hierarchijos viršūnėje būna išskirtinis/singuliarinis objektas, pavyzdžiui, objektas *Application* yra hierarchijos *Access Application* pradžioje, o *DBEngine* - hierarchijos *Jet* pradžioje.

3.3. Objektinis modelis *Access Application*

Trumpai apžvelgsime objektinio modelio *Access Application* objektus ir jų savybes. Hierarchijos pradžia – *Application*.

Objektas – *Application* - išreiškia pačią duomenų bazių valdymo sistemą *Access*, taip pat makrokomandų ir procedūrų *VBA* vykdymo terpę. Keičiant objekto *Application* savybes, keičiasi ir visos *Access* terpės charakteristikos. Makrokomandoms ir *VBA* programoms savybių sąrašas ir savybių aprašai yra kompaktiniame diske (*KD*), pateikiami lentelėse: `\MSO-97\Lenteles\Chapter3.pdf`. Savybių (*properties*) pavyzdžiai: *CurrentObjectName*, *MenuBar*, *UserControl* ir kt. TABLE 3.1: *Properties of Application object available in macro and VBA programming*.

(Savybių rusišką interpretaciją galima rasti knygoje: S.Novalis. *Access 2000 „LORY“*. Maskva. 2001 m.)

Objektų visybės *Forms* ir *Reports*. Pirmą kartą atveriant duomenų bazę sistema *Access* sukuria dvi visybes: *Forms*, atviroms formoms ir *Reports*, atviroms ataskaitoms. Atveriant formas arba ataskaitas *Access* atnauja tas visybes (*Forms* ir *Reports* visybių savybės žr. `\MSO-97\Lenteles\Chapter3.pdf`. TABLE 3.2: *Properties of the Forms and Reports collection objects*.

OBJEKTŲ VISYBIŲ SAVYBĖS (PROPERTIES COLLECTION OBJECTS).

Objektas *Form* – yra atvertoji forma ir kartu visybės *Forms* narys. Yra daugiau nei 100 objekto *Form* savybių, apibūdinančių formos išorę bei elgseną. Iš savybių sąrašo galima nustatyti 75 savybes: (žr. 30 svarbiausių savybių `\MSO-97\Lenteles\Chapter3.pdf`). TABLE 3.3: *Properties of the Form object available in both macro and VBA programming*. Pavyzdžiai: *ActiveControl*, *Application*, *Count*, *CurrentRecord*, *CurrentView* ir kt.

Objektas *Report* – tai atvertoji ataskaita ir kartu visybės *Records* narys. Ataskaitos savybių sąraše galima nustatyti 50 savybių.

Lentelėje TABLE 3.4: *Properties of Report object available in both macro and VBA programming* (`\MSO-97\Lenteles\Chapter3.pdf`) pateiktos svarbiausios savybės. Pavyzdžiai: *Application*, *Count*, *Filter*, *FilterOn*, *HasData* ir kt.

Savybės tik dėl *VBA* yra pateiktos: `\MSO-97\Lenteles\Chapter10.pdf` lentelėje TABLE 10.8 *Additional properties of the Report object*, o objekto *Report* metodai lentelėje TABLE 10.9 *Methodes of the Report object*. Metodų pavyzdžiai; *Circle* (*Draws a circle, an ellipse or an arc on a report when the Print event occurs*, brėžia apskritimą, elipsę arba lanką ataskaitoje, kai atpažįstamas įvykis *Print*); *Line*, linija (nubrėžiama linija); *Pset*, taškas (dedamas taškas) ir kt.

Objektų visybė Controls. Kiekvienai formai arba ataskaitai yra būdinga valdomųjų elementų visybė *Controls*. Savybės: žr. \MSO-97\Lenteles\Chapter3.pdf, lentelė TABLE 3.5: *Properties of the Controls collection object*. Pavyzdžiui: *Application*, *Count*.

Objektas Control yra formos arba apskaitos valdomasis elementas/valdomasis dėmuo ir kartu priklauso duotosios formos arba ataskaitos visybei. Valdomųjų elementų (priemonių skydeliui) *Toolbox* priklauso 17 įterptinių elementų. Įterptiniai valdomieji elementai (*built-in controls*), jų nustatytosios savybės, jų saitai su duomenimis, kitu objektu arba vaizdu yra pateikti: \MSO-97\Lenteles\Chapter3.pdf, lentelė TABLE 3.6: *The built-in controls*. Jų svarbiausios savybės, tipas ir leistinosios reikšmės: TABLE 3.7: *Core properties of controls available in macro and VBA programming*.

Kai kurios svarbiausios savybės, vartojamos programose, yra pateiktos: TABLE 3.8: *Some control properties useful in programming*.

Objektas Screen. Objektas *Screen*, ekranas, remiasi konkrečia, aktyvia forma, ataskaita arba aktyviu valdomuoju elementu. Panaudojant objektą *Screen* procedūroje *VBA* galima kreiptis į aktyvų objektą nežinant net jo vardo, tačiau pats kreipinys į objektą *Screen* formos, ataskaitos arba valdomojo elemento neaktyvina. Savybės objekto *Screen* yra pateiktos: \MSO-97\Lenteles\Chapter3.pdf, lentelė TABLE 3.9: *Properties of the Screen object*.

3.22 UŽDUOTIS.

Iš kompaktinio disko įkrauti lentelių rinkmeną *Chapte3.pdf* ir peržiūrėti lenteles, kuriose vardinamos objektų savybės.

Objektai Application, prieinami tik VBA. Objektinio modelio *Access Application* keletas objektų yra prieinami tik programuojant *VBA* kalba. Tai objektai *DoCmd*, *Module* ir visybė *Modules*, objektas *Reference* ir visybė *References*. Apie juos vėliau 10 skyriuje.

3.4. Objektinis modelis Access Visual Basic

Objektinis modelis *Access Visual Basic* apima tris objektus: *Debug*, *Err* ir *Collection*. Makroprogramavimo atveju panaudojama tik vieno objekto *Debug* savybė – *Immediate Window* (*immediate* – tiesioginis, betarpiškas). Išsamiau apie *Debug*, *Err* ir *Collection* žiūrėti 10 skyriuje.

Objektas Debug. Objektas *Debug*, derinamas, yra ypatingas objektas, prieinamas rengiant makroprogramas ir *VBA* programas. Jį galima panaudoti duomenims išvesti lange *Debug*. Specialios paskirties langas *Debug* pasirodo ekrane po komandos *Ctrl+G*. (Aplinkoje *Access 97* turi dvi sritis: apatinę sritį, vadinamą *Immediate*, ir viršutinę, kurioje galima atverti langus *Locals* ir *Watch*.)

Sritį *Immediate* galima panaudoti išraiškoms vaizduoti, savybėms teikti ir peržiūrėti, makrokomandoms ir procedūroms *VBA* vykdyti. Langas *Locals* tinka *VBA* procedūrų kintamųjų reikšmėms peržiūrėti. *Watch* leidžia peržiūrėti išraiškų ir kintamųjų reikšmes vykdant procedūras. Langai *Locals* ir *Watch* prieinami tik *VBA*.

Objektas *Debug* savybių neturi. Yra tik vienas metodas *Print*. Kreipinio į metodą *Print* sintaksė:
Debug.Print išvedimo-sąrašas Dirbant lange *Immediate* kreipinį galima paprastinti:
Print išvedimo-sąrašas (čia *išvedimo-sąrašas* skaitinė arba eilutės tipo išraiška arba išraiškos, atskirtos tarpeliais arba kabliataškiais).

3.5. Makroprogramavimo aplinka

Makrokomandomis yra rašomos instrukcijos betarpiškai *Access* sąsajai. *Jet* priemonės prieinamos be tiesioginės kreipties per duomenų bazės valdymo priemonės. Makroprogramavimo galimybės ribotos nes galima panaudoti tik įterptąsias savybes, o metodai nenaudojami.

Makroprogramavimo atveju galima naudoti tokius objektų šaltinius:

- Objektinio modelio *Access* tris visybes – *Forms*, *Reports*, *Controls*. Į aktyvų objektą kreipiamasi per objektą *Screen*. Lango *Debug* sritis *Immediate* panaudojama kreipinių ir išraiškų vaizdavimui, apskaičiavimui taip pat ir makroprogramoms vykdyti.
- Kreiptis į duomenis vykdoma panaudojant objektus *Table* kalba *DAO* (*TableDef*, *QueryDef*) ir *Field*. Kitų kreipties į duomenis objektų savybės nustatomos iš savybių sąrašų ir dialogų languose.
- Duomenų bazės lango makrokomandas.
- Dialogo langą *Customize*, kuris panaudojamas specializuoti komandų sąrašams, mygtukų deriniams bei greitosioms komandoms parengti.

3.5.1. Kreiptis į objektus, jų savybes, vardus

Objektą galima nurodyti makroprogramoje arba *VBA* procedūroje tik tinkamai jį identifikavus pagal *Access* reikalavimus.

Nuoroda/kreiptis – *reference*, *access* – kurio nors objekto nurodymas; simbolis, tekstas arba santrumpa, kuria į ką nors nurodoma.

Į objektą galima kreiptis keturiais būdais. Makrokomandose taikomas tik vienos krypties būdas – betarpiškas kreipimasis į objektą pagal jo vardą. Kiti būdai nagrinėjami 10 skyriuje.

Nuoroda į objektą pagal vardą. Nuoroda į objektą-taikinį yra rašoma objektų vardų maršruto eilute. Pirmasis objekto vardas, rašomas maršruto eilutėje, - hierarchijos pradžioje esančio objekto vardas, o paskutinis vardas – objekto taikinio vardas. Objektų vardai eilutėje rašomi be tarpelių, atskiriami šauktuku ir tašku. Pereinant iš objektų visybės į objektą dedamas šauktukas, pvz.: *Forms!formname*, o pereinant iš objekto į objektų visybę dedamas taškas, pvz.: *Forms!formname.Controls*, pvz.: *Forms!formname.Controls.controlname*.

Nuoroda į formą arba ataskaitą. Kaip taikomojoje programoje *Expenses* užrašyti betarpišką nuorodą į atvertą formą *Switchboard*? Reikia pradėti objektu *Application*, nurodyti visybę *Forms* ir objektą forma *switchboard*: *Application.Forms!Switchboard*.

Nuoroda į programos *Expenses* ataskaitą:

Application.Reports![Expense Report]

Kai objekto vardas parašytas ne vienu žodžiu (tarp vardo žodžių yra tarpeliai; pvz., *Expense Report*) objekto vardą reikia apskliausti laužtiniais skliaustais. Nors tai nėra būtina, sistema *Access* skliaustus gali padėti pati.

Vardų eilutės maršrutą galima trumpinti pagal nutylėjimą. Pvz.: yra žinoma, kad kreiptis į objektus galima tik tada, kai jau paleista *Access* sistema. Todėl aukščiau parašytas nuorodas galima sutrumpinti: *Forms!Switchboard*; *Reports![Expense Report]*

Nuoroda į formos arba ataskaitos savybę. Objekto savybė nuo objekto vardo atskiriama tašku. Tokios nuorodos sintaksė: *objekto-varas.savybės-varas*, pavyzdžiui: *Forms![Expense Categories].RecordSource*

Nors savybių sąrašė savybės vardas *Record Source* yra parašytas dviem žodžiais, nuorodos į savybę eilutėje, tarpelį tarp vardo žodžių galima praleisti ir rašyti – *RecordSource*.

PRATIMAS. SAVYBĖS REIKŠMIŲ NUSTATYMAS LANGE *IMMEDIATE*.

1. Darbo aplanke, pvz.: *C:\My Documents\Macro&VBA* sukurkite naują 3-čiojo skyriaus aplanką *PIRMADALIS-3sk*. Iš ankstesnio aplanko *PIRMADALIS-2sk* kopijuokite į naują aplanką *DB*, su kuria jau dirbote anksčiau, *DB_EXPENSES*. Įkraukite iš naujo aplanko *PIRMADALIS-3sk*

duomenų bazę *DB_EXPENSES* ir atlikote įvairius tyrinėjimus. (Tada *DB_EXPENSES*, kurią tyrinėjote 2 skyriuje išliks 2-jo skyriaus kataloge *PIRMADALIS-2sk* nepakitusi.)

2. Uždarykite duomenų bazės *DB_EXPENSES* komutacinio skydelio langą *Main Switchboard*, atverkite pagrindinį duomenų bazės langą *DB_EXPENSES*.

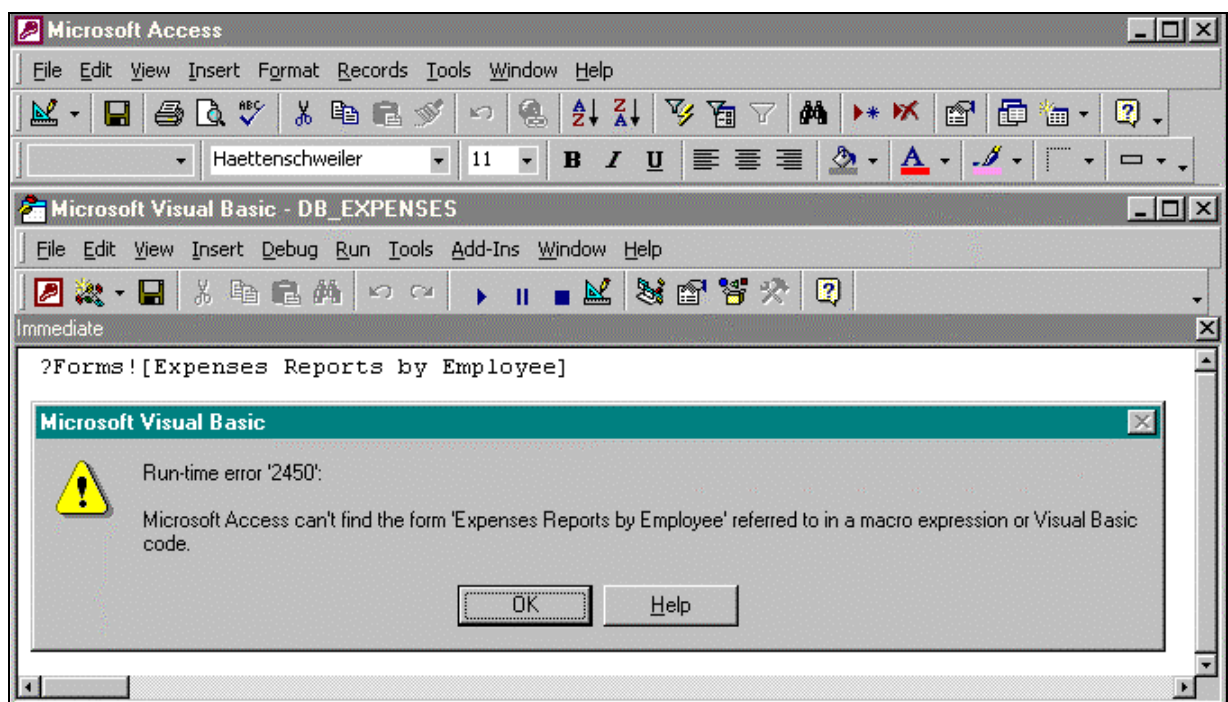
Duomenų bazės lange *DB_EXPENSE:Database*, atverkite formos *Expense Categories* vaizdavimo langą (mygtukas *Open*) ir surinkite greitąją komandą *Ctrl + G*. Ekrane pamatysite programavimo aplinkos *Microsoft Visual Basic* derinimo langą *Immediate*. Atverkite langą *Watch*, stebėti, ir *Local*, vietiniai (*View → Watch Window*, ir *Local Window*). Patogiau surikiuokite langus, tik pažiūrėkite ir pašalinkite langus *Watch* ir *Local*. Jų dabar nereikia.

3. Dialogo lango *Immediate*, tiesioginis/betarpiškas, surinkite žodį *Print* arba ? (klaustuką) ir nuorodą į pasirinkamą savybę. Po to spustelėkite *Enter*. Sistema *Access*, jeigu nebus klaidos, apskaičiuos/išves savybės reikšmę.

Jeigu įvesdami nuorodą padarėte klaidą, kai spustelėsite *Enter* gali pasirodyti pranešimas: *Run-time error '2465': Application-defined or object-defined error* (Klaida. Programos arba objekto nuorodos klaida). Tada *Debug Window* lango, srityje *Immediate* pasitikrinkite nuorodą. Klaidų atveju gali būti kitokių pranešimų, pavyzdžiui: *Run-time error '2450': Microsoft Access can't find the form 'Expense Reports' referred to in a macro expression or Visual Basic code. The form you referenced maybe closed or may not exist in this database. Microsoft Access may have encountered compile error in a Visual.*

Pamatykime pranešimą, kad yra klaida: sąmoningai apsirinkite, surinkime išraišką, kurioje nuoroda ne į formos savybę, bet į formos vardą:

?Forms![Expense Reports by Employee] ir spustelkit *Enter*. Matysite pranešimą, paskaitykite ir bakstelkit *OK* arba *Help*. Išbandykite abu pasirinkimus. Po to grįžkite vėl į langą *Immediate*.

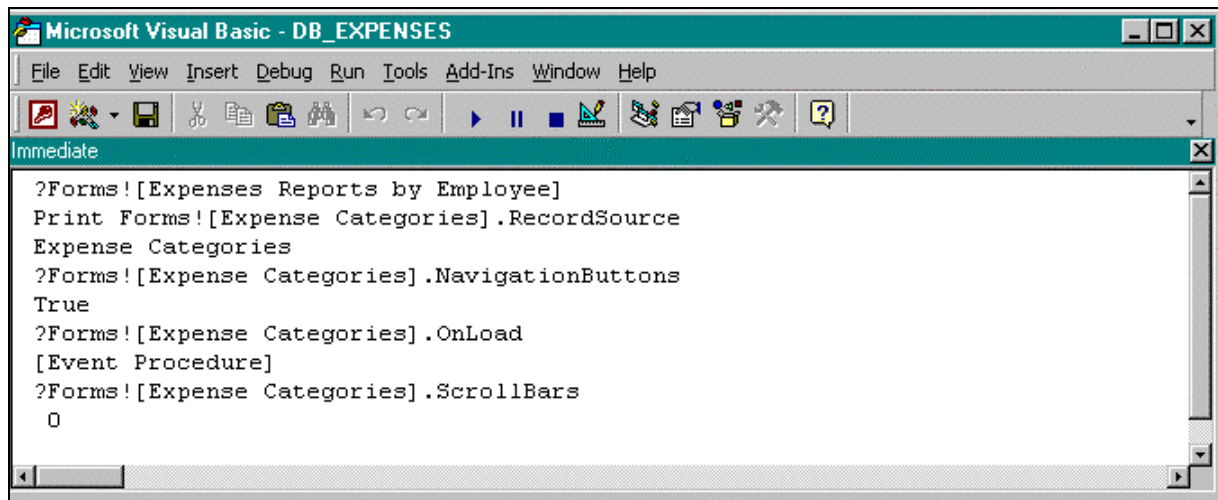


3.30 pav. Kai *Immediate* lange įrašoma išraiška, kurios reikšmės apskaičiuoti negalima, sistema praneša, kad yra vykdymo klaida ir pateikia apytikrį klaidos aprašą

Kitoje eilutėje surinkite nuorodą į atvertos formos *Expense Categories* savybę *RecordSource: Print Forms![Expense Categories].RecordSource*

Kai spustelėsite *Enter*, sistema *Access* eilutėje žemiau išves savybės *RecordSource* reikšmę *Expense Categories*.

Žemiau, atskiriomis eilutėmis, surinkite nuorodas į tos pačios formos savybes: *NavigationButtons*, *OnLoad*, *ScrollBars* ir kai po kiekvienos išvestos eilutės, išreiškiančios nuorodas, spustelėkite *Enter*, matysite savybių reikšmes. Įvesdami nuorodas naudokite komandas *Copy* ir *Paste* arba pažymėkite ir vilkite į naują eilutę. Galimos ir kitos žinomos *Edit* komandos.



3.31 pav. Objektų savybių reikšmių apskaičiavimo lange *Immediate* rodinys

[įveskite: *?Forms![Expense Categories].NavigationButtons*

?Forms![Expense Categories].OnLoad

?Forms![Expense Categories].ScrollBars

Turėtų būti gautos tokios reikšmės: *True*, *[Event Procedure]*, *0*.

Įveskite dar vieną nuorodą į kitos visybės *Reports*, dar neatvertos ataskaitos, savybę *Caption*: *?Reports![Expense Rpt Summary by Category].Caption*

Kai dabar spustelėsite *Enter*, ekrane pamatysite panašų ar kiek kitokį pranešimą: Run-time error: the report name 'Expense Rpt Summary by Category' you entered is misspelled or refers to a report that isn't open or doesn't exist. (Klaida: Įvedėte arba neteisingą ataskaitos vardą, arba tokios ataskaitos nėra).

Uždarykite *Microsoft Visual Basic* langą, uždarykite formos *Expense Categories* langą ir atverkite ataskaitos *Expense Rpt Summary by Category* langą (*Preview*). Ataskaitos lango *Expense Rpt Summary by Category* įvesties eilutėje *Beginning Rpt Date*, ataskaitos pradžios data, įveskite, pavyzdžiui, 95 04 01, o *Ending Rpt Date*, ataskaitos pabaigos data, įveskite 95 04 05, ir bakstelėkite formoje esantį mygtuką *Preview*, peržiūra. Ekrane matysite ataskaitos langą *Expense Rpt Summary by Category*. Surinkite komandą *Ctrl + G*, iškviestite ankstesnį langą *Immediate*. Nukelkite žymeklį į paskutinės nuorodos eilutės pabaigą (po žodžio *Caption*) ir spustelėkite *Enter*. Dabar jau matysite savybės *Caption* reikšmę: *Expense Rpt Summary by Category*.

Pakartokite tuos pačius veiksmus su ataskaitą *Expense Rpt Summary by Employee*. Lange *Immediate*, jeigu reikia, įveskite ataskaitos savybės *Caption* išraišką, jos reikšmei apskaičiuoti.

4. Padarykite išvadas.

5. Uždarykite langą *Immediate*, ataskaitų langus, grįšite į duomenų bazės langą *DB_EXPENSE:Database* Baikite darbą.

Nuoroda į valdomąjį elementą. Nuorodos eilutę reikia pradėti kaip įprasta objektu *Application* ir baigti valdomuoju elementu. Kai sistema *Access* jau paleista veikti, kaip prisimenate vardas *Application* praleidžiamas. Nuorodos sintaksė: *Forms!formas-vardas.Controls!valdomojo-elemento-vardas*; Pavyzdys: *Forms!Switchboard.Controls!Option1*

Objektams *Form* ir *Report*, nurodomiems visybėmis, valdomojo elemento visybę nurodyti nebūtina, nes sistema ją įveda pati pagal nutylėjimą. Tada nuorodos eilutė į valdomąjį elementą sutrumpėja: *Forms!Switchboard!Option1*

Nuoroda į valdomojo elemento savybę. Nuorodos sintaksė:

Forms!formas-varidas.Controls!valdomojo-elemento-varidas.savybės-varidas

Pavyzdys: norėdami kreiptis į formos *Expense Categories* tekstinio bloko *ExpenseCategory* duomenis panaudokite savybę *Value* štai taip:

Forms![Expense Categories]!ExpenseCategory.Value

Objektas – tekstinis sąrašas/blokas pagal nutylėjimą turi savybę *Value*, todėl ją išraiškoje galima neberodyti: *Forms![Expense Categories]!ExpenseCategory*

Pastaba: tekstinio sąrašo/bloko savybė *Text* taip pat nurodo valdomajam elementui tekstą. Skirtumas tarp *Value* ir *Text* yra tik toks, kad *Value* nurodo paskutinę išsaugotą lauko reikšmę, o *Text* dabartinę. Jeigu redaguojamas tekstas neišsaugomas, tai *Text* ir *Value* savybių reikšmės bus skirtingos. Kai įvestas tekstas, pvz., įrašas išsaugomas, neiškeliant žymiklio į kitą eilutę, valdomojo elemento reikšmė yra atnaujinama, tada reikšmės *Text* ir *Value* bus vienodos.

PRATIMAS. TEKSTINIO VALDOMOJO ELEMENTO SAVYBIŲ REIKŠMIŲ TEXT IR VALUE TYRIMAS.

1. Įkraukite duomenų bazės langą *DB_EXPENSE:Database*.

Atverkite formos *Expense Categories* vaizdavimo langą. Pirmojo įrašo reikšmę *Meals* įvesties eilutėje *Expense Category* pakeiskite į *Meal*. Neįrašykite pakeitimo.

2. Iškviaskite *Immediate* langą: *Ctrl+G*.

3. Lange *Immediate* surinkite nuorodas valdomojo elemento lauko *Expense Category* savybių charakteristikų reikšmėms apskaičiuoti. Įvedę kiekvienos nuorodos eilutę spustelėkite *Enter*.

?Forms![Expense Categories]!ExpenseCategory

?Forms![Expense Categories]!ExpenseCategory.Text

?Forms![Expense Categories]!ExpenseCategory.Value

Kai įvesite pirmąją nuorodos eilutę ir spustelėsite *Enter* sistema *Access* naujoje eilutėje išves reikšmę *Meals*, po antrosios nuorodos - reikšmę *Meal*, o po trečiosios nuorodos- *Meals*.

4. Padarykite išvadas ir baikite darbą: uždarykite langą *Immediate*. Baikite darbą su *Access*.

Kitos objektų savybės. Daugelį savybių tekstines reikšmes galima matyti lango srityje *Immediate*. Tačiau kai kurios savybės pasižymi savitomis reikšmėmis, kurios naudojamos nurodyti kitus objektus. Pavyzdžiui, valdomojo elemento savybė *Parent* nurodo vyresnįjį valdomąjį elementą, o tekstinio sąrašo/bloko savybė *Parent* nurodo formą, kurioje tas sąrašas yra.

Objektų kreipties į kitus objektus savybių sąrašas yra pateiktas kompaktiniame diske: Diskas \MSO-97\Lenteles\Chapter3.pdf, lentelė *TABLE 3.10: Object properties that refer to other objects*. Lentelėje pateiktos savybės *Me* ir *RecordsetClone* prieinamos tik *VBA* procedūrose. Išsamiau apie tai 10 skyriuje.

Kadangi lentelėje 3.10 pateiktomis savybėmis kreipiamasi į objektus, o ne į reikšmes, jų negalima patikrinti lango *Immediate* srityje. Pavyzdžiui, tekstinio bloko savybę *Parent* galima panaudoti nuorodai į pačią formą, tačiau jeigu surinksite išraišką: *?Forms![Expense Categories]!ExpenseCategory.Parent* ir spustelėkite *Enter*, *Access* praneš, kad tai klaida.

3.5.2. Nuorodos į pagalbinę formą

Pagalbinė forma yra pagrindinės formos dalis.

PRATIMAS.

Duomenų bazės *DB_EXPENSES* pagrindinė forma *Expense Reports by Employee* turi pagalbinę formą *Employees Subform*. Panagrinėkime kreipties į pagalbinę formą *Employees Subform* galimybes.

1. Atverkite iš 3 skyriaus aplanko, pvz.: PIRMADALIS-3sk duomenų bazės *DB_EXPENSES* pagrindinės formos *Expense Reports by Employee* vaizdavimo langą (*Open*).

2. Išveskite derinimo langą *Immediate* surinkite *Ctrl+G*.

3. Derinimo lango srityje *Immediate* galėsite sužinoti pagalbinės formos vardą, kuris išvedamas kaip pagalbinės formos valdomasis elementas, jeigu pateiksite nuorodą ir savybę *SourceObject* (*source* - šaltinis):

?Forms![Expense Reports by Employee]![Employees Subform].SourceObject ir spustelėsite *Enter*. Taip ir darykite.

Jeigu pasirodė pranešimas: *Run-time error'2465' Microsoft Access can't find the field 'Employees' referred to in your expression. You may have misspelled the field name, or the field may have been renamed or deleted. (Nerandamas išraiškoje nurodytas laukas 'Employees'. Galbūt neteisingai parašytas lauko vardas arba laukas pervardintas arba pašalintas)*. Bakstelėkite pranešimo lange *OK* ir ieškokite klaidos. Greičiausiai neapskliautėte *Employees Subform*. Apskliauskite ir vėl spustelėkite *Enter*. Žemiau nuorodos eilutės pamatysite reikšmę - *Employees Subform*.

4. Valdomojo elemento-pagalbinės formos *Employees Subform* savybė *Form* leidžia kreiptis į išvestą ekrane pagrindinę formą. Nurodykite savybės *Form* reikšmę *DefaultView*:

?Forms![Expense Reports by Employee]![Employees Subform].Form.DefaultView ir spustelėkite *Enter*.

Access išves skaičių 2, rodantį pateiktą atitinkantį lentelę *Datasheet*.

Tokią pateiktą matysite, kai uždarysite *Immediate* langą ir atvertos pagrindinės formos lange išskleisite mygtukų *View* sąrašą ir bakstelėsite *Datasheet View*. Matysite pagrindinės formos išraiškų lentelę.

5. Vėl grįžkite į *Immediate* langą. Užrašykite nuorodą į pagalbinėje formoje išvedamą valdomojo elemento savybę *Total Expenses*, kai spustelėsite *Enter*, sužinosite reikšmę 464: *?Forms![Expense Reports by Employee]![Employees Subform].Form.Controls![Total Expenses]*.

6. Suprastinkime paskutinę nuorodą. Pagalbinės formos visybę *Controls* sistema *Access* interpretuoja nutylėjimą, panašiai taip pat ir pagalbinės formos valdomąjį elementą *Form*.

Patikrinkite. Surinkite vieną ir kitą suprastintos nuorodos reikšmei *Total Expenses* rasti: vieną nuorodą be *Controls*. Galutinė nuoroda:

Forms![Expense Reports by Employee]![Employees Subform]![Total Expenses]

Turite gauti tą pačią reikšmę- 464.

Jeigu apsiriksate, gali pasirodyti, pavyzdžiui, toks pranešimas: *Run- time error '438'. Object doesn't support this property or method* (Objektui nebūdinga tokia savybė arba metodas). Tada ieškokite klaidos.

Vadinasi, nuorodos sintaksė į papildomosios formos valdomąjį elementą yra tokia: *Forms!pagr-form-varđ!papildomoji-forma-varđas!val-elem-varđ*

7. Perskaitykite aukščiau parašytas nuorodas, išvardinkite visybes, objektus, valdomuosius elementus, savybes. Išvardinkite duomenų bazės *DB_EXPENSES* konkrečius objektus ir elementus.

3.5.3. Nuorodos į veikiamųjų objektų valdomuosius elementus

Veikiamasis - *active* – programa, dokumentas, visas arba dalis displejaus ekrano tuo momentu atliekantis tam tikrą darbą.

Pradžioje nagrinėjome išsamias nuorodas arba identifikatorius pagal maršrutus nuo objektų hierarchijos pradžios iki paties objekto. Vėliau atsisakėme objekto *Application* ir sistemos *Access* pagal nutylėjimą interpretuojamų kai kurių visybių elementų.

Norėdami nurodyti veikiamosios formos arba ataskaitos valdomuosius elementus nuorodos eilutę taip pat galima trumpinti. Galima nerašyti veikiamosios/veiklios/aktyvios formos arba ataskaitos vardo. Pavyzdžiui, jeigu formą *Expense Reports by Employee* aktyvi, tai tekstinį bloką *FirstName* galima identifikuoti paprastai: *FirstName*, arba išvedama papildomoje formoje *Employees Subform* valdomąjį elementą *Total Expenses* nurodyti: *[Employees Subform]![Total Expenses]* Toks neakivaizdus veikiamosios/veiklios formos arba ataskaitos identifikatorius vadinamas sintaksės trumpiniu arba nepatikslinktu identifikatoriumi. Galima vartoti ir išbaigta patikslinkta nuoroda (kaip aiškinta anksčiau), tačiau pasitaiko atvejų, kai būtinas leistinas sintaksės trumpinys. Pavyzdžiui, panaudojant makroveiksmą *GoToControl*, kai reikia nurodyti elemento vardą veikliajame objekte, kuriam turi būti perduotas valdymas.

Derinimo lango srityje *Immediate* patikrinti identifikatorių į veikiamąjį objektą negalima, nes tuo atveju yra veikiamasis pats objektas – *Immediate* langas. Jeigu į langą *Immediate* įvesite nepatikslinktą nuorodą, *Access* praneš, kad kompiliavimo klaida: *Compile error. External name not defined* (Išorinis vardas nerastas).

Kompiliuoti – *compile* – pertvarkyti aukšto lygmens kalba parašytos programos pirminį kodą į objektinį kodą.

3.5.4. Objekto *Screen* naudojimas nuorodai į veikiamąjį objektą

Sistema *Access* turi unikalų būdą identifikuoti veikiamąją formą, ataskaitą, valdomąjį elementą, be to, elementą, kuris buvo veiklus paskutinis, net nevartojant specialių objektų vardų. Veikliajam objektui identifikuoti naudojamos objekto *Screen*, ekranas, savybės.

Pavyzdžiui, nuorodą į veikiamosios formos savybę *RecordSource*, įrašo šaltinis, galima surinkti taip: *Screen.ActiveForm.RecordSource*

Nuorodą į veikiamosios formos valdomojo elemento *LastName*, pavardę, savybę *Locked*, blokuotas, galima užrašyti taip: *Screen.ActiveForm!LastName.Locked*

Be to, veikiamosios formos veiklųjį valdomąjį elementą galima nurodyti: *Screen.ActiveControl*

Veiklaus valdomojo elemento vardą galėsite nurodyti: *Screen.ActiveControl.Name*

Veikiamosios formos tik ką veikliu buvusio valdomojo elemento savybę *TabIndex* pakartotinai galėsite nurodyti: *Screen.PreviousControl.TabIndex*

Nuorodų *Screen* lange *Immediate* patikrinti negalima, nes langas yra veikiamasis.

3.5.5. Nuorodos į lauką

Dažnai reikia kreiptis į lentelės arba užklauso lauką. Nuorodos sintaksei įtaką turi lentelės arba užklauso priklausymas atviros formos kaip pagrindiniam objektui, t.y. atviros formos įrašų šaltiniui:

Nuoroda į formos įrašų šaltinio lauką. Galima kreiptis į lentelės arba užklauso lauką, esantį formos įrašų šaltiniu. Sintaksė tokia: *Forms!formas-varas!lauko-varas*

1 PRATIMAS.

1. Duomenų bazės (iš aplanko PIRMADALIS-3sk) DB_EXPENSES lentelę *Employees*, darbuotojai, papildykite lauku *DateHired*, įdarbinimo data, ir įveskite lauko reikšmes kiekvienam darbuotojui. Į formą *Expense Reports by Employee* lauko, kitaip sakytume - valdomojo elemento, įjungti nereikia.

Pastaba: jeigu jau toks laukas -*DateHired* lentelėje *Employees* yra, papildykite kitu lauku – Priimtas dirbti. Įrašų lentelėje pirmajam darbuotojui parašykite datą, pavyzdžiui: 2000 05 01, antrajam 2000 06 05, trečiajam 2000 07 15. (*Short Date* ir *Input Mask*, kaukė, 0000.99.00;0)

2. Atverkite formą *Expense Reports by Employee*. Iškvieskite derinimo langą *Immediate* surinkite nuorodą: *?Forms![Expense Reports by Employee]!DateHired* ir spustelėkite *Enter*.

Jeigu lauko vardą keitėte, tai lauko vardo *DateHired* vietoj reikia įrašyti lauko vardą tokį, kokį įvedėte lentelėje *Employees*, pavyzdžiui: Priimtas dirbti.

3. Ekrano *Immediate* srityje matysite lauko *DateHired* arba 'Priimtas dirbti' pirmojo darbuotojo reikšmę.

4. Minimizuokite langą *Immediate* formoje *Expense Reports by Employee* nustatykite kitą įrašą, išskleiskite langą *Immediate*, nustatykite įvesties žymiklį į anksčiau surinktos nuorodos pabaigą ir spustelėkite *Enter*. Kartokite tuos pačius veiksmus dar kartą. Dabar lango *Immediate* srityje matysite visas tris trijų tarnautojų įdarbinimo lauko reikšmes, nors atvertoje formoje tokio lauko (valdomojo elemento) nėra.

Nuoroda į lentelės arba užklauso lauką. Konstruojant užklausą *SQL* kalba tenka kreiptis į lentelę arba kitą užklausą. Objektas *Field* visada pagal nutylėjimą priklauso visybei *Fields* todėl galima taikyti nuorodos sintaksę: *Lent-varas arba užkl-varas.Fields!lauko-varas ar lent-varas!* arba *užkl-varas!lauko-varas*

Lentelės ir užklauso valdo duomenų bazės branduolys *Jet*. Todėl pereinant iš visybės į visybės narį vartojamas taškas arba šauktukas. Pavyzdžiui, nuorodoje į lentelės *Employees* lauką *LastName* galima rašyti: *Employees!LastName* arba *Employees.LastName*

Lango srityje *Immediate* betarpiškai nuorodų į lentelę arba užklausą demonstruoti negalima. Tačiau galima panaudoti funkciją *DLookup()*.

2 PRATIMAS.

Derinimo lango srityje *Immediate* panaudoti funkciją *DLookup()* lentelės *Employees* pirmojo įrašo lauko reikšmę išvesti.

1. Atverskite lentelę *Employees (Open)* ir atverkite langą (Ctrl + G).

2. Lango srityje *Immediate* surinkite nuorodą *?DLookup("LastName","Employees")* ir spustelėkite *Enter*.

3. Ekrane, kitoje eilutėje bus išvesta pirmojo darbuotojo pavardė *Davalio*.

Apie funkciją *Dlookup()* aiškinama 4 skyriuje. Apie nuorodas į lenteles ir užklauso - 11 skyriuje.

3.6. Nuorodų rengimas komponuokliu *Expression Builder*

Sistemos *Access* išraiškos komponuokis/dariklis *Expression Builder* yra priemonė įvairioms nuorodoms rengti automatizuotai. Tačiau jo panaudoti derinimo lange *Immediate* negalima.

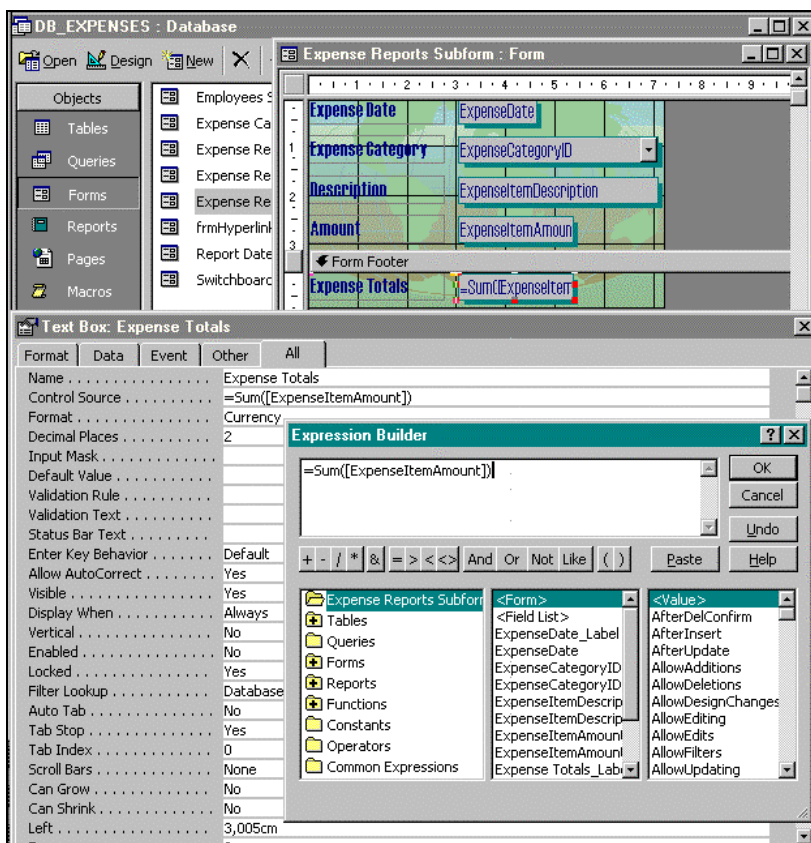
Expression Builder galima paleisti veikti ten kur reikia iš parankinio komandų sąrašo (reikia spustelėti pelės dešinįjį klavišą). Arba mygtukų juostoje bakstelėti mygtuką *Build*. Savybių parinkimo arba makroveiksmų redagavimo languose pakanka bakstelėti mygtuką esantį už įvesties laukelio pažymėtą trims taškais (pvz.: savybių *Properties* lange savybės *Control Source* laukelyje).

Expression Builder – kontekstinė priemonė, priklauso nuo to kur ji paleidžiama veikti.

Expression Builder langą sudaro dvi sritys: viršutinė konteksto ir nuorodos formavimo sritis ir apatinė, kurią sudaro trys blokai. Kairiajame bloke nagrinėjama duomenų bazės, visų lentelių, užklausų, formų ir ataskaitų aplankai ir įterpinių funkcijų, konstantų, operacijų dažniausių išraiškų ir specialiųjų *Visual Basic* funkcijų aplankai. Aplankai pažymėti pliuso ženklu yra sudėtiniai. Jų sudėtyje yra dar kiti aplankai. Kai atversite aplanką *Forms* arba *Reports*, matysite atverstų formų *Loaded Forms* arba atverstų aplankų *Loaded Reports* aplankus, taip pat ir aplankus formų su papildomomis formomis.

Kai kairiajame bloke pažymėsite objektą, pvz.: *Expenses Subform*, viduriniame bloke matysite objektus ir elementus, priklausančius pažymėtajam objektui. Kai centriniame bloke pažymėsite pasirinktą elementą, pav.: *TotalExpenses*, dešiniajame bloke matysite savybes.

Nuorodos išraiška (eilutė) formuojama lango *Expression Builder* viršutinėje srityje. Kai pažymėsite pasirinktą konkretų objektą, o vėliau objektą ar elementą, po to savybę į nuorodos eilutę perkelsite bakstelėdami mygtuką *Paste*. Operacijų ženklus nurodysite lango srityje esančiais mygtukais arba klaviatūros klavišais.



3.32 pav. Įrašų komponuoklio *Expression Builder* rodinys.

3.23 UŽDUOTIS.

1. Žiūrėdami į 3.32 pav. suraskite tinkamas komandas ir išveskite tokį pat *Expression Builder* rodinį.
2. Panašiai išveskite įrašų komponuoklį/dariklį pagrindinės formos bet kuriam laisvai pasirinktamam valdomajam elementui.

1 PRATIMAS.

Panaudodami *Expression Builder* surinkite patikslintą nuorodą į duomenų bazės *DB_EXPENSES* papildomos formos valdomąjį elementą *Total Expenses*.

1. Duomenų bazės *DB_EXPENSES* komutacijos lentelės langą *Main Switchboard* palikite atvertą arba uždarykite ir atverkite pagrindinį duomenų bazės langą.
2. Atverkite pagrindinę formą *Expense Reports by Employee* konstravimo lange, pažymėkite ir bakstelkite mygtuką *Build*. Pasirodys dialogo langas *Choose Builder*. Tačiau kartais *Expression Builder* galima iškviesti kitaip – gali tekti pasinaudoti savybių sąrašo langu (pvz., žr. 2.32 pav.).
4. Pasirinkite *Expression Builder*.
5. Apžvelkite dialogo langą *Expression Builder*, atverskite aplanką *Forms*, po to *Loaded Forms* – matysite atverstos formos aplanką *Expense Reports by Employee* ir formą komutacinės lentelės *Switchboard*. Atverskite formos aplanką *Expense Reports by Employee* – matysite pagalbinę formą *Employees Subform*.
6. Lango *Expression Builder* kairiajame bloke pažymėkite pagalbinės formos vardą *Employees Subform*, o centriniame bloke pažymėkite objektą *Total Expenses* ir viršutinėje lango srityje bakstelėkite mygtuką *Paste*.
7. Lango viršutinėje srityje, aukščiau operacijų mygtukų eilės, matysite surinktą patikslintą nuorodą (patikslintą identifikatorių): *Forms![Expense Reports by Employee]![Employees Subform].Form![Total Expenses]*
8. Uždarykite langą *Expression Builder* ir visus kitus langus, išskyrus duomenų bazės *DB_EXPENSES* langą.

2 PRATIMAS.

Kadangi komponuoklis *Expression Builder* yra kontekstinis, labai svarbu jo paleidimo veikti vieta. Priklausomai nuo paleidimo vietos galima formuoti kitokią nuorodą. Suformuokite nepatikslintą identifikatorių į objektą *Total Expenses*.

1. Duomenų bazės lange *DB_EXPENSES: Database* atverkite formos *Expense Reports by Employee* konstravimo langą (*Design*).
2. Mygtukų juostoje bakstelėkite mygtuką *Build*, dialogo lange *Choose Builder* nurodykite *Expression Builder*.
3. Apžvelkite *Expression Builder* langą. Pastebėkite pakitimus. Kadangi pagrindinė forma *Expense Reports by Employee* jau pažymėta kairiajame bloke, atverskite ją ir pažymėkite *Employees Subform*. Po to centriniame bloke pažymėkite *Total Expenses* ir bakstelėkite mygtuką *Paste*.
4. Dabar lango viršutinėje srityje matysite nepatikslintą nuorodą: *[Employees Subform].Form![Total Expenses]*
5. Uždarykite visus langus išskyrus duomenų bazės pagrindinį langą.

Komponuoklis *Expression Builder* gali būti panaudojamas išraiškoms, su specialiomis funkcijomis, parengti.

3 PRATIMAS. *EXPRESSION BUILDER* LANGE ATVAIZDUOTI DUOMENŲ BAZĖS *DB_EXPENSE* SPECIALIĄ FUNKCIJĄ IR FUNKCIJOS SINTAKSĘ.

1. Duomenų bazės *DB_EXPENSE* lange atverskite kortelę *Forms*, pažymėkite bet kurios formos vardą ir bakstelėkite mygtukų juostoje mygtuką *Code*.
2. Formos modulio lange iš parankinio komandų sąrašo nurodykite komandą *Build*.
3. Dialogo lange *Expression Builder* atverkite aplanką *Functions*, atverkite aplanką *DB_EXPENSE*. Centriniam bloke matysite *Global Code*, o dešiniajame specialią funkciją *IsLoaded*.
4. Bakstelėkite mygtuką *Paste*. Lango viršutinėje srityje matysite funkcijos *IsLoaded* iš standartinio modulio *Global Code* sintaksę: *IsLoaded*(«*strFormName*»)

3.7. Santrauka

- Objektų tarpusavio priklausomybė yra dvejopa: vieniems jų yra būdinga panašios/analogiškos savybės ir elgsena, kitus sieja vyresnumo/hierarchiniai ryšiai.
- Taikomąją programą *Microsoft Access* sudaro du svarbūs komponentai (kiekvieno jų sava hierarchija): *Access Application* ir duomenų bazės branduolys *Jet*.
- Bendruoju atveju kreipiantis į objektą makrokomandoje arba procedūroje *VBA* reikia vartoti patikslintą nuorodą. Tokia nuoroda pradedama hierarchijos pirmuoju objektu ir tęsiama visybėmis ir objektais iki objekto–taikinio. Ėjimas nuo objekto, į jo visybę žymimas operacijos ženklu tašku, o nuo visybės į vieną jo narį operacijos ženklu - šauktuku.
- Nuorodos gali būti trumpinamos atsižvelgiant į visybių ir savybių nurodymą pagal nutylėjimą.
- Nuorodoje į objekto savybę reikia nurodyti objektą ir savybės vardą, atskiriama nuo jo operacijos ženklu tašku.
- Objekto *Screen* savybė leidžia kreiptis į veiklųjį objektą, nenurodant jo vardo.
- Į formos įrašų šaltinio langą galima kreiptis net tada, kai formoje nėra siejančio su tuo lauku valdomojo elemento.
- Nuorodų ir išraiškų komponuoklį *Expression Builder* galima panaudoti patikslintiems ir nepatikslintiems nuorodoms rengti.

4. Veiksmai su formomis

- Valdomųjų elementų ir duomenų saitai
- Valdomųjų elementų sąveikavimo pagrindai
- Laisvas valdomasis elementas kaip kintamasis
- Kombinuotojo bloko panaudojimas informacijos paieškai
- Dviejų formų sinchronizavimas

Sistemos *Access* taikomosiose programose duomenų forma turi ypatingą paskirtį. Rengiant visiškai automatizuotą taikomąją programą, kurią panaudos kiti vartotojai, yra sudaroma specializuota vien tik formų ir ataskaitų sąsaja. Duomenų įvedimas ir jų peržiūra vykdoma panaudojant formas. Dažniausiai makrokomandos ir *VBA* procedūros yra lyg atsakas į formų teikiamus įvykius.

Šiame knygos skyriuje iš esmės nagrinėjamos dvi pagrindinės temos: saitai tarp formų ir duomenų šaltinių lentelių ir santykiai tarp formų valdomųjų elementų. Be to, čia apžvelgiami svarbūs formų projektavimo klausimai:

- Taikymas užklausų formuojamų priemone *AutoLookup*, informacijai gauti iš lentelių ir automatiškai duomenų paieškai lentelėje.
- Užklausų apskaičiuojamųjų laukų panaudojimas.
- Formoje apskaičiavimams atlikti valdomieji elementai taikomi informacijai gauti iš tos pačios arba kitų formų, lentelių laukų ir užklausų panaudojant domeno agregatinę funkciją *DLookup*.
- Laisvo formos valdomojo elemento taikymas laikinai saugoti reikšmę.
- Lentelių arba užklausų kombinuotųjų blokų panaudojimas duomenims įvesti.
- Dviejų formų sinchronizavimas taikant įterptinę papildomąją formą.
- Dviejų formų, esančių skirtinguose languose, sinchronizavimas, parengiant procedūras paslaugų programa *Form Wizard*.

Domenas – *domain* – duomenų su išskirtiniais parametrais sritis duomenų aibėje.

Agregatas – *aggregate* - visuma, kad ir nevienarūšių objektų, kuri tam tikra, pvz., loginių ryšių, prasme yra bendra.

4.1. Saitai tarp formų ir duomenų valdymo elementų

Jeigu santykinė/reliacinė duomenų bazė (tokie yra *Ms Access DB*) suprojektuota teisingai, tai dubliuojama informacija sutinkama tik lentelių sietinuose laukuose. Visuose kituose laukuose informacija pateikiama tik vieną kartą. Kai duomuo, t.y. kurio nors lauko reikšmė lentelėje pakeičiama kita reikšme vienoje vietoje, pakeitimas automatiškai perkeliamas į visas formas, ataskaitas ir užklausas.

Dažnai tenka pateikti vienodų įrašų duomenims skirtinguose objektuose. Pavyzdžiui, kiekvienoje paraiškoje tenka minėti to paties paraiškėjo pavardę bei adresą.

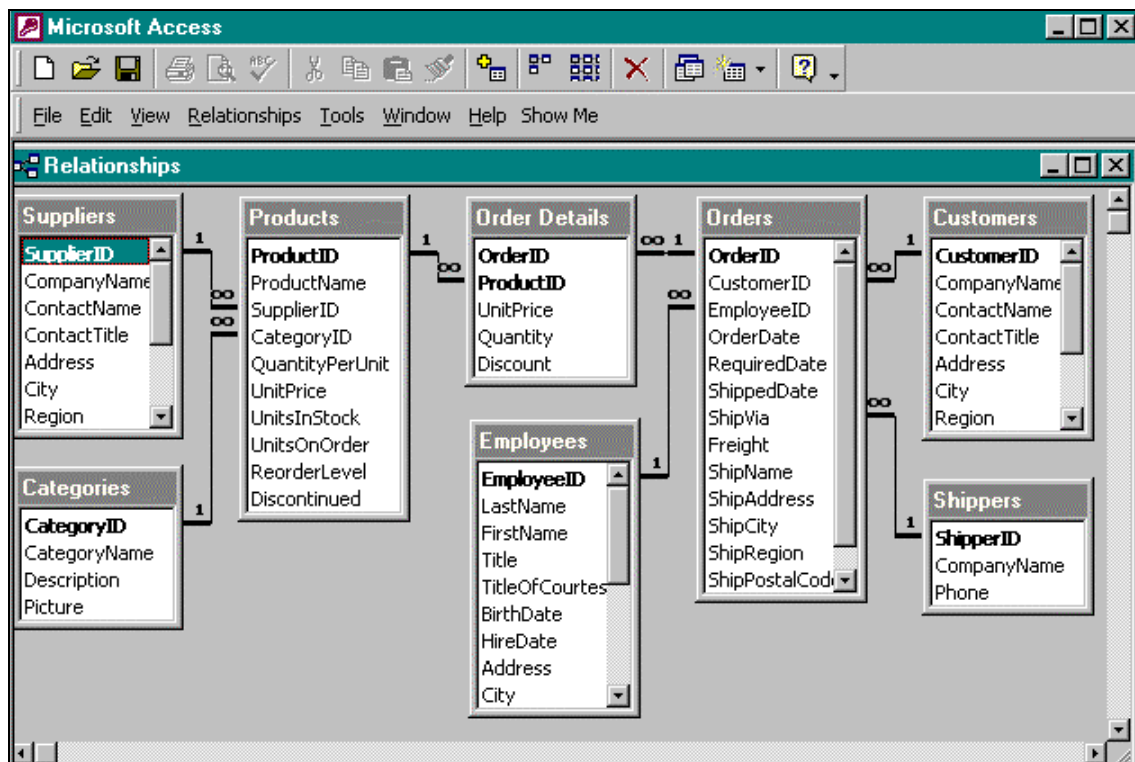
Sistema *Access* turi įvairias priemones, išgauti reikiamai informacijai iš įvairių lentelių ir nukreipti ją į kitą tinkamą objektą.

Šiame skyriuje atliksime daug pratimų ir užduočių su duomenų bazė pavyzdžio *Northwind Traders* duomenimis.

4.24 UŽDUOTIS.

Ankščiau parengtame savo darbo aplanke, pvz.: *C:\My Documents\Macro&VBA\PIRMADALIS-4sk* raskite (Start, Find Files or Folders, laukelyje Named įrašykite North* ir bakstelkite Find Now) įkelkite naujos duomenų bazės *Northwind.mdb* kopiją ir pavadinkite ją *Northwind_Sk4.mdb*. Atverkite rinkmeną *Northwind_Sk4.mdb*. Duomenų bazės lange *Northwind_Sk4:Database* įkelkite lentelių saito langą *Relationships* ir apžvelkite lentelių saitų schemą: Tools → Relationships arba bakstelėkite mygtukų juostoje atitinkamą mygtuką.

(Aiškinimą apie profesionaliai parengtą duomenų bazę *Northwind Traders* raskite ir paskaitykite konsultacijų žinyne *Help*).



4.33 pav. Lentelės ir jų saitai duomenų bazėje *Northwind*

4.1.1. Įrašų rinkinio samprata

Kai atveriamą lentelę, vykdoma ilgalaikėje atmintinėje saugoma užklausa arba vykdomas užklauskų struktūrinės kalbos operatorius *SQL*, duomenų bazės branduolys *Jet* operatyviojoje atmintinėje parengia įrašų rinkinį (*recordset*).

Įrašų rinkinys - tai grupė įrašų lentelėje arba aibė įrašų, gaunamų vykdant užklauską arba *SQL* operatorių. Ne visos užklauskos generuoja įrašus, pavyzdžiui, veiksmų užklauskos jokių įrašų nesukuria.

Kai kada reikalingi duomenys saugomi vienoje lentelėje, pvz., tarnautojų adresai lentelėje *Employees*, tarnautojai. Tačiau dažniausiai duomenys pasirenkami ne iš vienos lentelės. Tarkime, kad reikalingi adresai tarnautojų, aptarnaujančių tik tam tikrą šalį, pavyzdžiui, tiekiančių gaminius į Airiją (*Ireland*). Informacija apie gaminių tiekimą saugoma lentelėje *Orders*, paraiškos. Todėl tenka vieną informaciją gauti iš lentelės *Employees*, o kitą iš lentelės *Orders*. Labai svarbu užklausoje yra duomenų, saugomų ne vienoje lentelėje, vienijimas (*join*).

PRATIMAS.

Apžvelkime svarbiausius užklauskų parengimo veiksmus. DB *Northwind_4sk* iš lentelių *Employees* ir *Orders* parenkime naują užklauską *qryEmployeesShipCountry*, būdingą Airijai (*Ireland*).

Užklauskos grafinis projektavimas. Užklauskos konstravimo lange (*Design*) parenkite užklauską rengimo pagal pavyzdį *QBE* būdu (*QBE-query by example*). Į užklauskos konstravimo srities pirmąją ląstelę *Field* iš lentelės *Orders* tempkite lauką *ShipCountry*, gavėjo šalis. Iš lentelės *Employees* į *Field* eilutės kitas ląsteles tempkite laukus *FirstName*, *LastName* ir adreso laukus: *Address*, *City*, *Region*, *Country*.

Saito tarp lentelių nustatymas atliekamas lange *Relationship*. Saito linija, jungianti lentelių laukus *EmployeesID* informuoja apie įrašų jungimo būdą „vienas su daugeliu“ (prie lentelės

Employees ženklas 1, o prie lentelės *Orders* ∞). Pagal reikalą galima nustatyti kitoki jungimo būdą.

Apskaičiavimų vykdymas. Užklauso projektavimo srities eilutės *Field* tuščiose ląstelėse galima surinkti apskaičiavimų išraiškas. Vykdam užklausa gauta išraiškos apskaičiavimo reikšmė duomenų bazėje nesaugoma, skaičiuojama iš naujo kiekvieną kartą, kai vykdoma užklausa. Todėl ją galima tik matyti ir skaityti. Pakeiskite laukus *First Name* ir *Last Name* vienu lauku *Full*, visas, taikydami seku junginio išraiška.

Įterpkite *Field* eilutėje naują stulpelį prieš lauką *FirstName* ir laukų vardų eilutėje *Field* surinkite išraišką: *FullName:[LastName] & " , "& [FirstName]*

Jungimas - concatenation – keleto sekų (simbolių, žodžių, eilučių, rinkmenų) jungimas į vieną seką.

Panaikinkite buvusius stulpelius *FirstName* ir *LastNname*.

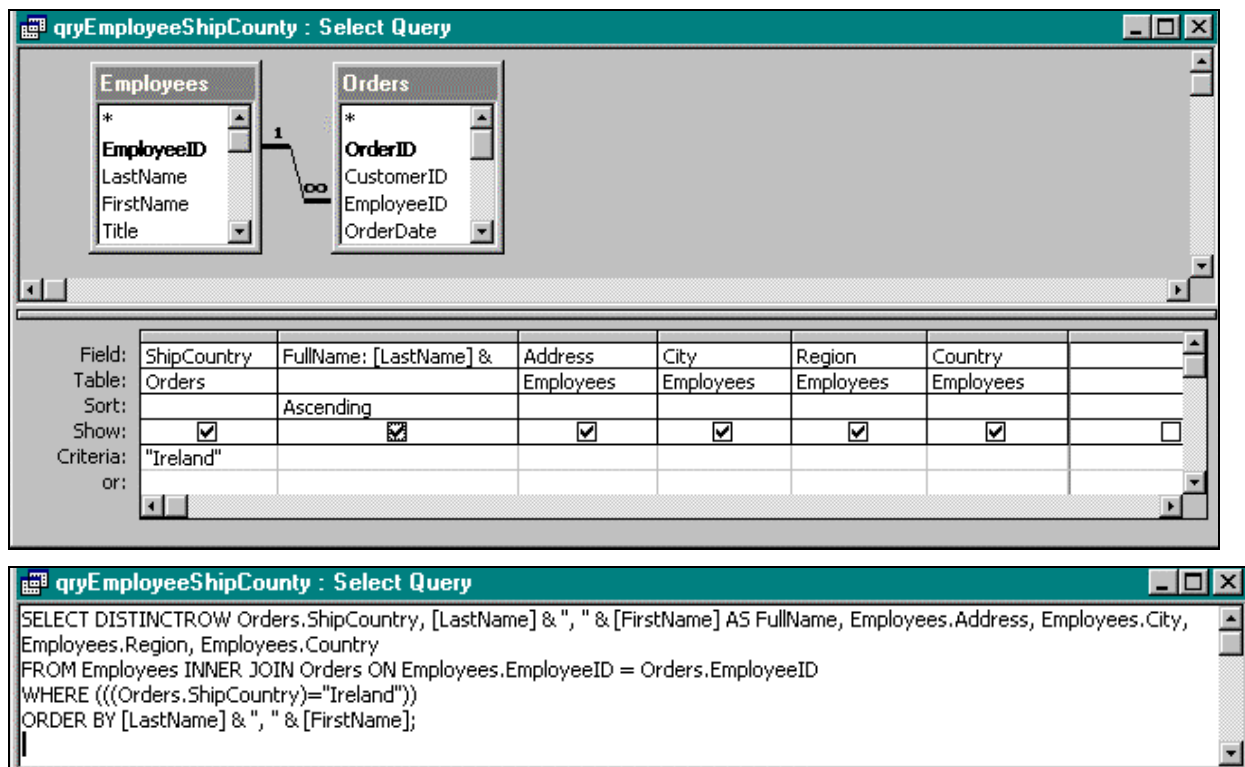
Įrašų ribojimas ir rikiavimas. Įrašų kiekis, kuris bus išvedamas po užklauso vykdymo, yra ribojamas eilutėse *Criteria* ir *Or* įrašant išraiškas. Įrašų rikiavimą vykdo eilutėje *Sort* nustatyta komanda. Stulpelio *ShipCountry*, šalis gavėja, eilutės *Criteria*, kriterijus, ląstelėje įrašykite šalies gavėjos Airijos pavadinimą *Ireland*. Lauko *FullName* eilutės *Sort* ląstelėje nustatykite komandą *Ascending*, pagal didėjimą.

Lygiavertis operatorius SQL. Kai konstruojama užklausa (*Design*) sistema *Access* automatiškai parengia užklausiai atitinkamą operatorių *SQL*, kurio veikimą (užklausa, parengtą *SQL* kalba) galima matyti: *View → SQL View* . Užklauso operatoriaus *SQL* galima redaguoti ir rengti tiesiogiai. Be to, specialiosios užklauso *Union* gali būti parengiamos tik taikant *SQL*. Tokia yra *DB Northwind* užklausa *Customers and Suppliers by City*, jungianti lenteles *Customers*, klientai, ir *Suppliers*, tiekėjai.

Peržiūrėkite dabar rengiamos išrankos (*Select*) užklauso *qryEmployeesShipCountry* rodinį *SQL*, paskaitykite ir bandykite suprasti, o kai baigsite darbą su šia užklausa, peržiūrėkite ir palyginkite *DB Northwind* suvienijimo (*Union*) užklauso *Customers and Suppliers by City* rodinį *SQL* ir *Datasheet View*.

4.25 UŽDUOTIS.

1. Kopijuokite iš kompaktinio disko į savo darbo aplanką atsisiųsta DB rinkmeną *Northwind_Ch4.mdb*, įkraukite DB ir peržiūrėkite parengtą iš vienos lentelės užklausa *qryEmployeeField*.
2. Išveskite ir perskaitykite *SQL* užklausa (*View → SQL View*).
3. Kopijuokite iš kompaktinio disko į savo darbo aplanką parengtą aplinkoje *Access97*, bet jau konvertuota į *Access 2000* DB rinkmeną *Ch4_North-konvert.mdb*.
4. Įkraukite DB ir peržiūrėkite parengtą iš dviejų lentelių užklausa *qryEmployeeShipCountry*.
5. Išveskite ir perskaitykite *SQL* užklausa.
6. Išveskite į ekraną užklausa parengtų įrašų rinkinį.
7. Palyginkite dabar matomą užklausa su savo parengtą užklausa DB *Northwind_4sk*, kurią teko rengti atliekant pratimą.



4.34 pav. Užklauso *qryEmployeeShipCountry* rodinys konstravimo aplinkoje *QBE* ir reiškiamas *SQL* kalba

Užklauso išsaugojimas. Nepriklausomai nuo to ar užklausa yra suformuota taikant *QBE* arba *SQL* ją galima išsaugoti dviem būdais:

- Užklausa išsaugoma, kaip duomenų bazės pagrindinis objektas *Queries*. Duomenų bazės branduolys *Jet* analizuoja išsaugotą užklausa ir parengia optimizuotą jos įvykdymo algoritmą, ir užklausa bus vykdoma pagal tą algoritmą.
- Sistema *Access* leidžia išsaugoti užklausa *SQL* kaip savybę kito objekto. Pavyzdžiui, jeigu forma parengiama pagal užklauso vykdomo rezultatų lentelę, tai operatorius *SQL* gali būti išsaugotas kaip duotosios formos savybė *RecordSource*. Kiekvieną kartą atveriant formą *DB* branduolys *Jet* analizuoja operatorių *SQL*, operatyviai parengia ir vykdo užklauso vykdomo algoritmą.

Išsaugokite užklausa ir įvykdysite ją – mygtukas *Run*. Matysite įrašų rinkinį.

Užklauso vykdomo rezultatas – įrašų rinkinys, kaip objektas, saugomas operatyviojoje atmintinėje. Uždariant užklausa įrašų rinkinys dingsta.

Užklauso vykdomo rezultatų lentelę galima išvesti įrašų lentelės *Datasheet* pateiktimi. *DB* branduolys valdo formų ir ataskaitų saitus su šaltiniais – lentelėmis ir užklausomis. Dirbant su *DB* interaktyviai (dialogo būdu) nėra reikalo žinoti apie objektų – šaltinių ir objektų taikinių saitus. Tačiau norint automatizuoti *DB* dažnai tenka rengti programas, galinčias keisti formoje arba ataskaitoje vaizduojamus duomenis, todėl reikia suprasti sąveikas „įrašų rinkinys – forma“ ir „įrašų rinkinys – ataskaita“. Tai labai svarbu.

Išveskite į ekraną lentelės įrašų būdu *Datasheet View* atmintinėje saugomą užklausa *qryEmployeeShipCountry* parengtų įrašų rinkinį.

Projektuose *Access* užklauso panaudoti negalima. Vietoj užklauso panaudojami tos pačios paskirties kiti du objektai: saugomos procedūros ir pateiktys/atvaizdavimo rodiniai. Saugoma procedūra (*stored procedure*) – tai parengtinai sukompiliuotų *SQL* operatorių rinkinys, turintis vieną vardą ir vykdomas kaip vientisas vienetas. Atvaizdavimo rodiniai (*view*) – tai pagal *DB* užklauso sukurtos virtualios duomenų lentelės. Jos padeda optimizuoti bendrą *Access* ir serverio *SQL Server* darbą.

4.1.2. Formos arba ataskaitos įrašų šaltinis

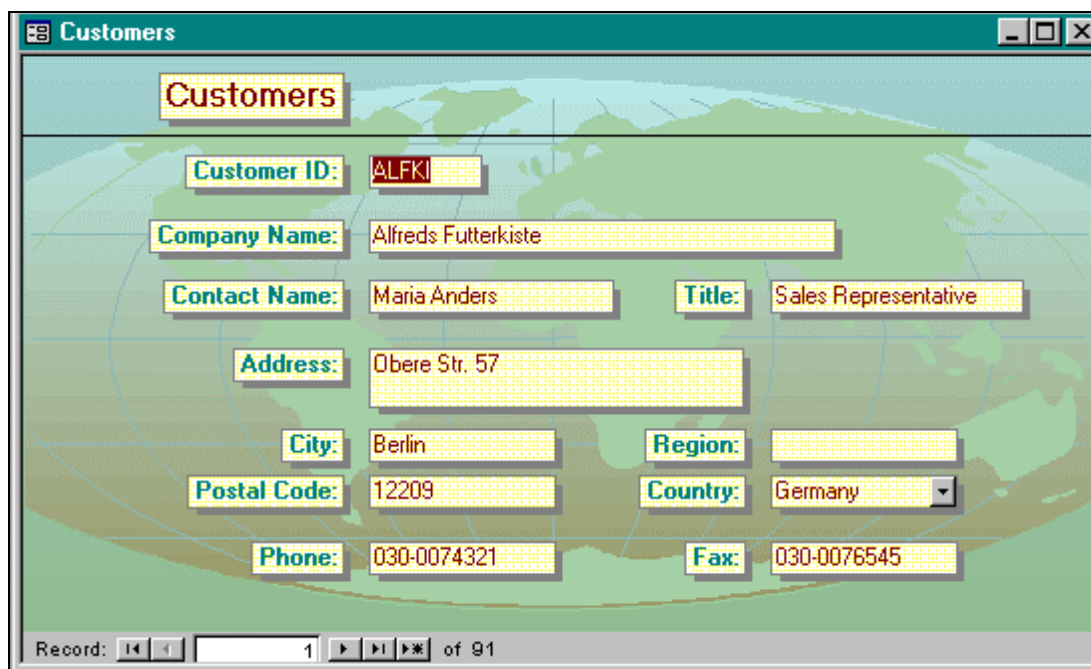
Formos ir ataskaitos gali būti *bound*, susietosios, ir *unbound*, nesusietosios. Rengiant formą arba ataskaitą duomenų šaltinis nurodomas savybe *RecordSource*, įrašų šaltinis. Jei ta savybė lieka tuščia, tai forma arba ataskaita bus nesusieta (*unbound*), t.y. saitas su įrašų rinkiniu nebus nurodytas.

Savybės *RecordSource* reikšmė gali būti lentelės, užklauskos arba generuojančio įrašus operatoriaus *SQL* vardas. Tuo atveju forma ar ataskaita bus susieta (*bound*) su lentele, užklausa arba operatoriaus *SQL*. Susietosios formos arba ataskaitos valdomieji elementai, veikia kaip atskiri langai, vaizduojantys įrašų rinkinio laukus. Atveriant susietąją formą arba ataskaitą *DB Jet* atmintinėje parengia įrašų rinkinį pagal savybės *RecordSource* reikšmę.

PRATIMAS.

Atverkite *DB Northwind Sk4* duomenų vaizdavimo formą *Customers*, klientai, susietą su tokio pat vardo lentele. Peržiūrėkite įrašų rinkinį panaudodami tyrinėjimo/judėjimo (*navigation*) mygtukus. Formos konstravimo lange atverkite savybių langą, pastebėkite savybės *RecordSource* reikšmę *Customers*.

Uždarant formą, formos įrašų rinkinys išnyksta, nes jis egzistavo tik operatyviojoje atmintinėje.



The screenshot shows a Microsoft Access form titled 'Customers'. The form has a light blue header with the title 'Customers'. Below the header, there are several text boxes for data entry, each with a label to its left. The labels and their corresponding values are: 'Customer ID: ALFKI', 'Company Name: Alfreds Futterkiste', 'Contact Name: Maria Anders', 'Title: Sales Representative', 'Address: Obere Str. 57', 'City: Berlin', 'Region: ', 'Postal Code: 12209', 'Country: Germany' (with a dropdown arrow), 'Phone: 030-0074321', and 'Fax: 030-0076545'. At the bottom of the form, there is a status bar that reads 'Record: 1 of 91'.

4.35 pav. Formos *Customers*, klientai, pirmojo įrašo rodinys

4.1.3. Valdomųjų elementų duomenų šaltiniai

Formoje arba ataskaitoje valdomieji elementai yra parengiami pagal savybės *ControlSource*, valdomojo elemento šaltinis reikšmę, taip pat gali būti susietieji arba nesusietieji.

Susietieji valdomieji elementai. Jeigu forma arba ataskaita yra susieta su duomenų šaltiniu, tai su juo galima sieti ir valdomąjį elementą. Toks valdomasis elementas vadinamas susietuoju (*bound*) ir veikia kaip lentelės duotasis laukas.

1 PRATIMAS.

Atverskite formos *Customers* konstravimo langą, raskite lauko reikšmių *CompanyName* savybės *ControlSource* reikšmę. Matysite valdomojo elemento šaltinio savybės *ControlSource* reikšmę - *CompanyName*. Tai susietos formos šaltinio lentelės lauko vardas, sutampantis su formos lauku.

Nesusietieji valdomieji elementai. Nesusietieji (*unbound*) valdomieji elementai (kurių savybės *ControlSource* reikšmė nenurodoma) būna dviejų rūšių: vietoj *ControlSource* reikšmės įvedama išraiška reikšmei nustatyti, arba reikšmės vieta pasilieka kaip ir buvo tuščia. Jeigu įvedama išraiška, tada toks valdomasis elementas vadinamas skaičiuojamuoju. Išraiška *ControlSource* įvedama makrokomanda arba *VBA* procedūra atveriant formą ir pagal ją parengto valdomojo elemento redaguoti klaviatūra negalima. Kitu atveju, kai *ControlSource* reikšmė nėra nurodyta, valdomąjį elementą galima įvesti arba redaguoti klaviatūra savo nuožiūra.

Parengto valdomojo elemento bet koku būdu reikšmė išlieka kol atvira duomenų forma arba ataskaita.

Valdomieji elementai su dviejų duomenų šaltiniais. Sistema *Access* leidžia siesti kombinuotąjį bloką/sąrašą (*Check Box*) arba paprastą bloką/sąrašą (*Text Box*) su duomenų šaltinio lauku arba palikti elementą nesusietą. Tie valdomieji elementai turi savybes ne tik valdomojo elemento šaltinį *ControlSource*, bet dar ir *RowSource*, eilutės šaltinį. Pastarąją savybę galima nurodyti šaltinį duomenų, išvedamų sąrašą eilutėse. Galimi keturių rūšių šaltiniai:

- lentelė, užklausa arba *SQL*, generuojanti duomenis
- vartotojo parengtų reikšmių sąrašas
- laukų vardų sąrašas lentelėje, užklausoje arba *SQL*
- reikšmių sąrašas, nustatomas specialios paskirties funkcija

Forma su teksto bloku arba kombinuotuoju bloku leidžia asociacijas su dviem įrašų rinkiniais: formos įrašų rinkiniu, pateikiamu pagal formos savybę *RecordSource* ir paties valdomojo elemento įrašų rinkiniu pagal jo savybę *RowSource*.

Pavyzys: Formos *Orders*, paraiškos, pagalbinė forma asocijuota su dviem įrašų rinkiniais: papildomosios formos įrašų rinkiniu sudaromu pagal užklausą *Order Details Extended*, paraiškų detalių papildymas, ir įrašų rinkiniu eilutėms kombinuotojo bloko *Product*, gaunamo iš lentelės *Products*, gaminiai. Peržiūrėkite pavyzdį.

Asociacija; sugretinimas - *association*- bendrais požymiais susietų objektų (duomenų, reiškinių, procesų, daiktų) visuma; loginė objektų sąsaja.

Valdomasis elementas be savybės *ControlSource*. Be savybės *ControlSource* gali būti dviejų rūšių valdomieji elementai: elementai visiškai nesiejami su duomenimis ir elementai koku būdu asocijuojami su duomenimis, bet neišvedantys konkrečių reikšmių. Pavyzdžiui, kai formoje įrengiamas nesiejamas jungiklio mygtukas, parinkčių jungiklis arba perjungimo mygtukas tai savybę *ControlSource* galima nurodyti valdomajam elementui duomenų šaltinį. Tačiau kiekvienas toks valdomasis elementas sujungtas į parinkčių grupę netenka savarankiškumo ir tampa grupės papildomuoju elementu. Kiekvieno papildomojo elemento savybė *ControlSource* pakeičiama savybe *OptionValue*, o savybė *ControlSource* priskiriama duomenų parinkčių grupei.

2. PRATIMAS

Atverkite formą *Orders* ir peržiūrėkite valdomojo elemento ir parinkčių grupės savybių langus. *Check Box* *United* (konstravimo lange ties valdomojo elemento *United* patikros jungiklio spustelkite pelės dešinį klavišą ir pasirinkite *Properties*) ir *Option Group: ShipVia* (bakstelkite du kartus patikros jungiklių *Speedy*, *United* ir *Federal* grupės apvado/rėmelio liniją).

4.36 pav. Sudėtinės formos *Orders*, paraiškos, rodinys

4.2. Šaltinio modelis „viena forma – vienas duomenų šaltinis“

Kai forma arba ataskaita turi tik vieną duomenų šaltinį, sistema *Access* naudoja šaltinio modelį „viena forma – vienas duomenų šaltinis“.

Jeigu reikia panaudoti dviejų susietų lentelių duomenis, tai įrašų šaltinis parengiamas užklausa arba *SQL* operatoriumi. Tokiu tikslu jau esame parengę užklausą *qryEmployeesShipCountry*. Jau žinote, kad formą galima parengti ne vienu, o dviem įrašų rinkiniais.

Sistema *Access* formos įrašų rinkinius gali parengti automatiškai panaudojant automatizuota duomenų keitimo priemonę – užklausą tipo *AutoLookup*.

3 PRATIMAS.

Tarkime, kad duomenų bazei *Northwind_sk4* reikia parengti formą *frmCustomersOrders* naujoms paraiškoms įvesti. Formoje turi būti pareiškėjo adresas ir paraišką apibūdinančios žinios. Užklausą reikia rengti iš dviejų lentelių *Customers*, klientai, ir *Orders*, paraiškos (4.2.1 skyr.). Formą konstruoti pagal užklaustos vykdymo rezultatų lentelę. Pareiškėjų adresus reikia įvesti automatiškai, o paraiškų duomenis turi įvesti vartotojas.

4.2.1. Automatinis duomenų keitiny *AutoLookup*

AutoLookup panaudojamas užklausiai formuoti iš dviejų lentelių, automatiškai užpildo lauką reikšmėmis iš vienos lentelės, kai tik įvedama naujam įrašui gauti bendrojo lauko (*join field*) kita reikšmė.

AutoLookup galės veikti, jeigu tenkinamos tokios keturios sąlygos:

- dviejų lentelių iš kurių sudaroma užklausa, įrašai turi būti siejami būdu „vienas su daugeliu“;
- po lentelių sujungimo vienos lentelės bendrasis laukas turi būti indeksuojamas (*Indexed*) reikšme *Yes (No Duplicate)*, nors to lauko pirminis raktas nėra būtinas; ta lentelė toliau bus vadinama pirmąja;

- užklausoje formuojamas bendrasis laukas turi būti imamas iš tarpusavyje santykiaujančių lentelių antrosios lentelės. Toks laukas pirmojoje lentelėje yra pagrindinis, o antrojoje lyg pagalbinis, tačiau būtinas saitui tarp pirmosios ir antrosios lentelių nustatyti. Pavyzdžiui į užklausą reikia įvesti lauką *CustomerID* iš lentelės *Orders*, bet ne iš lentelės *Customers*.
- į užklausos bendrąjį lauką įvedama reikšmė turi būti lygi vienos iš lentelių jau žinomai lauko reikšmei. Pavyzdžiui, mūsų užduotyje esančio kliento atpažiniklio/identifikatoriaus *CustomerID* reikšmę iš lentelės *Customers*.

PRATIMAS.

Jeigu tokios sąlygos yra tenkinamos, tai įvedus bendrojo lauko reikšmę sistema *Access* automatiškai ieško atitinkamo įrašo santykinėje lentelėje. Rengdami užklausą su automatinio duomenų keitimu *AutoLookup* įvykdykite šiuos žingsnius:

1. Įprastu būdu (*QBE*) parenkite užklausą iš lentelių *Customers*, klientai, ir *Orders*, paraiškos.
2. Į užklausos konstravimo sritį perkeltkite kai kuriuos lentelės *Customers* laukus, išskyrus lauką *CustomerID*.
3. Perkeltkite iš lentelės *Orders* kai kuriuos laukus, iš jų taip pat ir lauką *CustomerID*.
4. Atvaizduokite lentelę *Datasheet* ir lauko *Customers* kombinuotajame sąrašė *CustomerID* nurodykite klientą. Laukai iš lentelės *Customers* bus užpildoma automatiškai.
5. Įrašykite užklausą vardu *qryCustomersOrders*. *DB Northwind_sk4* forma *Orders* užklausos rezultatus panaudoja kaip antrąjį įrašų šaltinį.
6. Peržiūrėkite formą *Orders*. Kai formos kombinuotajame sąrašė *Bill To*, sąskaita, nurodykite klientą, matysite informacija automatiškai pakeičiama pagrindinės formos srityje. Be to, informacija automatiškai išvedama ir *ShipTo*, gauti/tiekti, skirsnyje, nors tai nėra automatinės paieškos rezultatas (apie tai vėliau).

AutoLookup – labai svarbi priemonė, padedanti konstruoti duomenis, įvedimo formas ir automatiškai ieškoti informacijos kitose lentelėse.

4.26 UŽDUOTIS.

1. Peržiūrėkite *DB Northwind_4sk* svarbesnes užklausas, pvz.: *Alphabetical List Products*, *Employee Sales by Country*, *Invoices*, *Order Subtotals*, *Products Above Average Price*, *Products by Category*, *Sales by Category*, *Ten most Expensive Products*.
2. Pastebėkite ir trumpai apibūdinkite kiekvienos užklausos ypatumus. Atskirai pasidomėkite kriterijais, skaičiuojamaisiais laukais, išvedamais rezultatais.

4.3. Valdomųjų elementų ir laukų sąveika

Kai formos įrašų šaltinis yra parengtas pagal užklausą su priemone *AutoLookup*, valdomasis elementas, susiejamas su bendruoju lauku (elementas *Join*) yra labai svarbus. Jeigu pakeisite to elemento reikšmę, tai sistema *Access* automatiškai modifikuos reikšmes valdomųjų elementų, siejamų su laukais *Lookup*, keitiniai. Vadinasi, tarp valdomojo elemento *Join* ir valdomųjų elementų *Lookup* nustatomas saitas.

Formos valdomasis elementas (formos laukas), siejamas su lentelės lauku sukuria dvikryptę komunikaciją. Lentelės lauko reikšmė rengiant formą yra perkeliama į formos lauką (formos valdomąjį elementą). Formos valdomojo elemento reikšmę formoje galima redaguoti, bet į lentelės lauką ta reikšmė bus perkelta tik tada, kai bus formoje išsaugota.

Formoje valdomieji elementai vieni kitiems informaciją gali perduoti dviem būdais: panaudojant skaičiuojamuosius laukus formos įrašų šaltinyje (pvz., užklausoje) ir panaudojant skaičiuojamuosius laukus pačioje formoje.

4.3.1. Užklauso skaičiuojamieji laukai

Skaičiuojamajame lauke išvedama apskaičiavimo reikšmė. Skaičiuojamojo lauko vardas rašomas išraiškos kairėje, atskiriamas dvitaškiu. Pagal nutylėjimą sistema *Access* išveda vardą *ExprN*, čia N - skaičius, išraiškos eilės numeris. Išraiškoje galima nurodyti operacijas, funkcijas, nuorodas į kitus užklauso laukus arba nuorodas į reikšmes, grąžinamas kitų reikšmių tipo objektų ir valdomųjų elementų ir atvirų formų.

PRATIMAS.

Reikia parengti užklausą iš lentelės *Employees*. Panaudokite skaičiuojamąjį lauką tarnautojų pavardėms ir vardams sujungti. Parenkite pagal užklausą ir išsaugokite formą.

1. Renkite įprastą užklausą *SelectQuery (QBE)*, įkelkite visus lentelės *Employees* laukus.
2. Po lauko *ReportsTo* įterpkite skaičiuojamojo lauko stulpelį eilutės *Field* ląstelėje surinkite papildomą vardą ir išraišką: *FullName: LastName & ", " & FirstName*

Išsaugokite užklausą vardu *qryEmployeeField*.

3. Sukonstruokite formą pagal parengtą užklausą panaudodami paslaugų programą *AutoFormWizard (New, AutoForm: Columnar, qryEmployeeField, OK)*.
4. Peržiūrėkite parengtą formą. Įsitikinkite, kad valdomojo elemento *FullName* reikšmės modifikuoti negalima.
5. Pakeiskite valdomojo elemento *FirstName* reikšmę ir klavišu *Tab* arba pele perkeltkite žymiklį į valdomąjį elementą *FullName*. Pastebėkite, kad *FullName* reikšmė pasikeičia automatiškai.
6. Įrašykite formą vardu *frmEmployeeField*.

4.4. Formos skaičiuojamieji laukai

Formos skaičiuojamieji valdomieji elementai išveda išraiškų rezultatus. Išraiška įvedama savybės *ControlSource* įvesties lauke ir pradedama ženkle "lygu". Išraiškoje gali būti nurodomi tokie pat dėmesys kaip ir užklauso skaičiuojamojo lauko išraiškoje. Duotojo elemento savybei *Name* galima suteikti vardą bet kurio kito formos valdomojo elemento.

4.27 UŽDUOTIS.

Sukonstruokite formą su skaičiuojamuoju lauku iš lentelės *Employees*.

1. Konstruokite formą panaudodami *AutoFormatWizard*.
2. Formos konstravimo lange (*Design View*), žemiau lauko *ReportsTo* įterpkite nesąryšinį (*unbound*) lauką *FullName* ir jo savybės *ControlSource* įvesties lauke surinkite išraišką: *=LastName & ", " & FirstName*
3. Įsitikinkite, kad skaičiuojamojo valdomojo elemento (lauko) *FullName* reikšmės pakeisti negalima.
4. Pakeiskite reikšmę lauko *FirstName* ir *Tab* klavišu žymiklį perkeltkite į kitą valdomąjį elementą. Valdomojo elemento *FullName* reikšmę dabar matysite pakeistą.
5. Grąžinkite tikrąją reikšmę ir įrašykite vardu *frmEmployeesControl*. Nesietinis apskaičiavimų valdomasis elementas turi tik vienkryptį saitą, pvz.: iš lentelės lauko į

formos skaičiuojamąjį elementą. Išraiška pateikiama savybės *ControlSource* įvesties lauke, gali gauti duomenis iš trijų šaltinių:

- iš tos pačios formos;
- iš kitos atviros formos;
- iš kito duomenų šaltinio panaudojant funkciją *DLookup()*.

4.4.1. Kaip gauti duomenis iš tos pačios formos

Formos laisvas valdomasis elementas duomenis gali gauti iš kitų tos pačios formos valdomųjų elementų. Nuorodoje į lauką arba valdomąjį elementą galima vartoti sintaksės trumpinį – nepatikslintąjį nuorodą. Taip sistemai *Access* nurodoma, kad laukas yra duomenų šaltinyje arba toje pačioje formoje.

Skaičiuojamųjų valdomųjų elementų išraiškų, gaunančių duomenis iš tos pačios formos, pavyzdžiai yra pateikti *KD* diske: *\MSO-97\Lenteles\Chapter4.pdf TABLE4.1:ControlSource expressions*

Pastaba. Kai valdomajame elemente vartojamas agregatinės funkcijos *Sum()*, *Avg()* arba *Count()*, galima kreiptis į formos duomenų šaltinio lauką. Tuo atveju reikia nurodyti lauko vardą, o ne formos valdomąjį elementą. Jeigu gaunate duomenis iš elemento, siejamo su lauku, bet pavadinto kitu vardu (nesutampančiu su lauko vardu), tai elementas išveda *#Name?* pranešdamas, kad klaida. Pavyzdžiui, kai reikia apskaičiuoti reikšmių sumą formos valdomajame elemente *GrandTotal*, kuris siejamas su formos šaltinio lentelės lauku *Amount*, įveskite išraišką *=Sum(Amount)*, o ne *=Sum(GrandTotal)*.

Objekto *Screen* panaudojimas. Norint išvesti dabartinio veiklaus valdomojo elemento įrašo vardą nesietiniame valdomajame elemente savybei *ControlSource* reikia suteikti reikšmę tokios išraiškos: *=Screen.ActiveForm.CurrentRecord*

Pereinant į kitą įrašą sistema *Access* automatiškai vykdo perskaičiavimus ir nesietiniuose elementuose išveda naujas reikšmes.

PRATIMAS. PATYRINĖKITE OBJEKTO *SCREEN* NAUDOJIMĄ.

1. Į formą *frmEmployeesControl* žemiau lauko *FullName* įterpkite du teksto laukus (*Text Box*) su savybėmis *Name* ir *ControlSource*:

<i>Name</i>	<i>ControlSource</i>
<i>txtPrevious</i>	<i>=Screen.PreviousControl</i>
<i>txtActive</i>	<i>=Screen.ActiveControl</i>

2. Perjungiant į formos vaizdą *Form View* valdomasis elementas *txtActive* išveda veikliąją elemento *EmployeeID* reikšmę 1. Valdomasis elementas *txtPrevious* pateikia reikšmę *#Error* todėl, kad atveriant formą nėra (nebuvo anksčiau) veiklaus elemento.

3. Klavišu *Tab* pereikite į kitą formos valdomąjį elementą ir nedarykite jokių pakeitimų. Nei *txtActive*, nei *txtPrevious* automatiškai neatnaujinami.

4. Vykdykite priverstinį valdomąjį elementų atnaujinimą komanda *Records→Refresh* arba spustelėkite funkcinį klavišą *F9*.

5. Pažingsniuokite: *Tab* po to *F9* ir t.t. Jeigu reikia, redaguokite, t.y. pakeiskite valdomojo elemento *LastName* reikšmę ir klavišu *Tab* pereikite į kitą valdomąjį elementą. Apskaičiavimų valdomieji elementai *FullName* ir *txtPrevious* atnaujinami automatiškai.

4.28 UŽDUOTIS.

1. Iš kompaktinio disko įkraukite atsisiųsta DB *Northwind_Ch4* ir išbandykite pagal pratimą papildytų valdomųjų elementų reikšmių keitimą.
2. Iš kompaktinio disko įkraukite konvertuota DB *Ch4_North-konvert*, išbandykite papildytų valdomųjų elementų veikimą, peržiūrėkite savybių reikšmes, pastebėkite skirtingą savybių žymėjimą. Atkreipkite dėmesį į formos kitus ypatumus, valdomųjų elementų kitokią išdėstymą.

Einamųjų/dabartinių duomenų išvedimas į ekraną. Tik ką nagrinėtame pavyzdyje nesiejamą valdomojo elemento reikšmė automatiškai neatnaujinama, kol nesuredaguojuamas formos valdomasis elementas. Interaktyvaus darbo metu apskaičiavimų valdomuosius elementus galima atnaujinti priverstinai spustelint klavišą *F9*. Automatiškai atnaujinti galima tik pagal parengtą duomenų atnaujinimo programą. Rengiant programą pirmiausia tenka nuspręsti, koks įvykis turėtų paleisti programą veikti.

Pereinant formoje į kita valdomąjį elementą buvęs elementas atpažįsta įvykius *Exit* ir *LostFocus*, o naujas elementas – įvykius *Enter* ir *GotFocus*. Programai paleisti pereinant iš vieno formos elemento į kitą galima panaudoti bet kurį įvykį. Pereinant skirtingose formose į kitus valdomuosius elementus pasirinkti tinkamus įvykius sudėtingiau todėl, kad ne kiekvienas valdomasis elementas atpažįsta visus tinkamus įvykius. (Apie įvykius buvo išsamiai aiškinta 3 skyriuje).

Einamųjų/dabartinių duomenų išvedimo ypatumai ir sužadinimas. Duomenys, atvaizduojami veikiamajame/veikliajame ekrane, kai kada gali būti ne patys paskutiniai, todėl kad:

- jei atversti du langai arba peržiūrinėjate duomenis daugelio vartotojų terpėje, tai pakeitimai viename lange ne visuomet automatiškai perkeliama į kitą langą;
- formos valdomojo elemento skaičiuojamojo lauko reikšmė automatiškai neperskaičiuojama, kol nepereisite į kitą įrašą;
- sąrašų blokuose suformuotuose priemonė *Lookup* iš kitos lentelės, duomenys automatiškai neatnaujinami, kai atliekami keitimai lentelėje;

Yra trys būdai duomenims ekrane sužadinti: atkūrimas, perskaičiavimas ir apklausa.

Atkūrimas – *refreshing* – pakeičia esamus įrašo duomenis.

Komanda: *Records* → *Refresh* arba funkcinis klavišas *F9*. Išvedami atnaujinti ir papildyti įrašai, šalinami pažymėti šalinimui įrašai.

Perskaičiavimas – *recolculating* – leidžia atkurti apskaičiavimų valdomuosius elementus spustelint klavišą *F9*.

Apklausa (*pooling*) pakartotinai vykdo objekto parengimo užklausą. Formos apklausa sužadinama komanda *Shift+F9*.

Jeigu formoje yra sąrašų blokai, kurių šaltinis yra ne toks pats, kaip formos duomenų šaltinis, tai formos apklausos komanda tokių sąrašų blokų šaltinio nevykdo.

PRATIMAS.

Patikrinkite duomenų bazės *Northwind_4sk* formų *Products*, gaminiai, ir *Categories*, kategorijos, elgseną. Formos *Products* duomenų šaltinis yra lentelė *Products*, o toje pačioje formoje *Category* kombinuotojo bloko – operatorius *SQL* su nuoroda į lentelę *Categories*.

1. Peržiūrėkite formų savybes. Atverkite abi formas formų vaizdavimo languose. Formą *Categories* papildykite savos redakcijos nauju įrašu. Įrašykite įrašą. Formos *Products* kombinuotame bloke *Category* naujos kategorijos nerasite. Tačiau, jeigu dirbate su *Access 2000*,

naujai įrašytą kategoriją rasite ir 2 po to 3 punkto vykdyti nebereikia. Naują įrašytą kategoriją galėsite pašalinti tik atvertę įrašų lentelę.

2. Surinkite komandą *Shift+F9* skirtą apklausti formos duomenų šaltinį, bet ne kombinuotojo bloko šaltinį.

3. Vykdykite formos *Products* interaktyvųjį atkūrimą pavartodami dvi komandas *F9*, o po to *Shift+F9*. Arba uždarykite ir vėl atidarykite formą *Products*.

4. Išbandykite komandas, stebėkite jų veikimą.

4.4.2. Kaip gauti duomenis iš kitos formos arba ataskaitos

Kita forma arba ataskaita, iš kurios norima gauti duomenis turi būti atvira. Vykdomojo elemento savybės *ControlSource* įvesties lauke įvedama patikslintoji nuoroda.

Pavyzdžiui, kai DB *Northwind_4sk* atvarta forma *Orders* ir forma *Employees*, formos *Orders* valdomojo elemento vardo savybėje *ControlSource* galima nurodyti tarnautojų vardus, kai surinksite tokią išraišką: *=Forms!Employees!LastName & ", "& Forms!Employees!FirstName*

Žemiau pateikiami formos *Orders* nesiejamo elemento savybės *ControlSource* nuorodų išraiškų reikšmių pavyzdžiai. Nuorodose kreipiamasi į atviros formos *Customer Orders* valdomuosius elementus:

- *=Forms![Customer Orders]! CustomerID* – lauko *CustomerID*, formos *Customer Orders* arba šaltinio duomenims į ekraną išvesti.
- *=Forms![Customer Orders]!Customer Orders Subform1]!OrderDate* - pagalbinės formos valdomojo elemento *OrderDate* einamajai įrašo reikšmei išvesti.
- *=Orders Subform]!UnitPrice* - valdomojo elemento *UnitPrice* einamajai reikšmei pagalbinėje formoje išvesti. Skaičiuojamasis elementas priklauso pagrindinei formai:
- *=Parent!OrderID* - parodo pagrindinės formos einamosios pagalbinės formos valdomojo elemento *OrderID* reikšmę. Skaičiuojamasis elementas priklauso pagalbinei formai.

PRATIMAS.

Išraiškose galima rašyti nuorodas į dvi atvertas formas. Pavyzdžiui, kai atvira forma *Customers* ir *Employees*:

1. Įterpkite į formą *Customers* nesiejamą valdomąjį elementą *txtBoth* (pvz., pavadinimo eilutėje *Customers*), kuris turi išvesti abiejų formų einamųjų įrašų numerius. Valdomojo elemento savybės *ControlSource* įvesties lange surinkite savybės visą išraišką vienoje eilutėje: *= "The Customers current record is " & Screen.ActiveForm.CurrentRecord & " and the Employees current record is " & Forms!Employees.CurrentRecord*

2. Peržiūrėkite formos *Customers* įrašus. Valdomas elementas atnaujinamas automatiškai.

3. Peržiūrėkite keletą formos *Employees* įrašų. Po to vėl grįžkite į formą *Customers*. Matysite, kad valdomasis elementas *txtBoth* dabar automatiškai neatnaujinamas.

4. Formoje *Customers* vykdykite elementų perskaičiavimą: *Records* → *Refresh* arba spustelėkite klavišą *F9*. Duomenų atkūrimo procesą galima automatizuoti. Pavyzdžiui, formą *Customers* galima sužadinti bakstelėdami formoje pele, tad galima pradėti rengti makrokomandą arba procedūrą *VBA*. Kai bakstelėjote formą pele, ji atpažįsta įvykį *Active*, kuris gali paleisti veikti proceso automatizavimo programą.

4.28 UŽDUOTIS.

1. Iš kompaktinio disko įkraukite atsisiųsta DB *Northwind_Ch4*, išbandykite ir patikrinkite pratimą.
2. Iš kompaktinio disko įkraukite konvertuota DB *Ch4_North-konvert*, išbandykite ir patikrinkite pratimą.
3. Peržiūrėkite savybių reikšmes, pastebėkite skirtinga savybių žymėjimą. Ilgą savybės išraišką patogų visą matyti lange *Zoom*. (Ties išraiškos eilute surinkite sparčiųjų klavišų kombinaciją *Shift + F2*).
4. Pastebėkite kiekvienos formos kitus ypatumus, valdomųjų elementų kitoki išdėstymą.

4.4.3. Duomenų gavimas iš kito šaltinio funkcija **DLookup()**

Kai kada reikia į formą įvesti duomenis iš lentelių arba užklausų, kurios nėra duotosios formos duomenų šaltiniai. Tuo atveju *ControlSource* savybės lauke naudojama funkcija *DLookup()*. Pavyzdžiui, kai į formą *Products* reikia įvesti naują gaminį (produktą), pageidautina jo kategorijos apibūdinimą gauti iš lentelės *Categories*. Tuo atveju į formą *Products* reikia įterpti nesiejamą (*unbound*) valdomąjį elementą ir jo savybėje *ControlSource* nurodyti funkciją *DLookup()*.

Domeno agregatinės funkcijos. *DLookup()* – domeno agregatinės funkcijos pavyzdys. Domenas - tai lentelė arba užklausa. Domeno agregatinės funkcijos yra šios:

- *DLookup()* – grąžina nurodyto lauko reikšmę.
- *DMin()*, *DMax()* – grąžina didžiausią arba mažiausią nurodyto lauko reikšmę.
- *DFirst()*, *DLast()* – grąžina nurodyto lauko reikšmę iš pirmojo arba paskutiniojo įrašo.
- *DAvg()* – grąžina nurodyto lauko aritmetinį vidurkį.
- *DSum()* - grąžina nurodyto lauko reikšmių sumą.
- *DStDev()*, *DStDevP()* - grąžina nurodyto lauko standartinį nuokrypį arba visybės standartinį nuokrypį.
- *DVar()*, *DVarP()* - grąžina nurodyto lauko standartinę dispersiją arba visybės dispersiją.
- *DCount()* - grąžina nurodyto lauko įrašų skaičių su nulinėmis reikšmėmis.

Funkcijos *DLookup()* sintaksė. Funkciją galima panaudoti gauti lentelės arba užklausos lauko reikšmei:

- Užklausoje. Apskaičiavimų išraiškoje eilutės *Field* ląstelėse, eilutės *Criteria* ląstelėse arba užklausos atnaujinimo ląstelės išraiškoje *Update To*.
- Makrokomandose. Sąlygoje arba veiksmo argumente.
- Procedūrose *VBA*. Sąlygoje arba argumentų metoduose.
- Formoje arba ataskaitoje. Apskaičiavimo valdomajame elemente.

Lentelė arba užklausa, kurioje funkcija vykdo paiešką, vadinama funkcijos domenu. Funkcijai priklauso trys argumentai. Pirmasis argumentas lauko vardas. Antrasis - lentelės arba užklausos vardas (funkcijos domeno vardas). Trečiasis - įrašo paieškos arba išrankos sąlyga.

Sintaksės bendras vaizdas: =DLookup(“lauko-var”, “lentelės/užklausos-var”, “paieškos-sąlyga”)

PRATIMAS. PASITYRINĖKITE FUNKCIJOS *DLOOKUP* VEIKIMĄ.

1. Į formą *Products* įterpkite nesiejamą valdomąjį elementą - teksto lauką vardu *txtDescription* (pvz., *Unit Price* eilutėje) ir savybės *ControlSource* įvesties lauke surinkite išraišką: *=DLookup(„Description“;„Categories“;„CategoryID = Forms!Products!CategoryID“)*

Šioje išraiškoje paieškos sąlyga nurodo išranką įrašų, kurių laukas *CategoryID* sutampa su reikšme formos valdomojo elemento *CategoryID*.

Paieškos sąlygai taikoma sintaksė: *lauko-varđ = Forms!formos-varđ!valdomojo-elemento-varđ*

Čia kairiojoje paieškos sąlygoje nurodomas lauko vardas lentelės arba užklausos, kurioje vykdoma paieška, o dešiniojoje – patikslintoji nuoroda į formos valdomąjį elementą, kuriame yra nurodyta ieškoma reikšmė.

2. Pereikite į kitą formos *Products* įrašą. Pastebėkite, kad valdomasis elementas *Category Description* atnaujinamas automatiškai.

3. Kombinuotojo bloko sąrašė nurodykite kitą kategoriją. Pastebėkite, kad nesietinis valdomasis elementas automatiškai neatnaujinamas.

4. Atkurkite/atnaujinkite patys (interaktyviai): *Records* → *Refresh* arba spustelėkite klavišą *F9*.

Domeno agregatinių funkcijų tyrimas. Visoms domeno funkcijoms tinka ta pati sintaksė. Trečiasis argumentas nėra būtinas. Jei trečiasis argumentas nenurodomas, tai funkcija nurodo visą įrašų rinkinį, kai įrašas nenurodomas arba domenas neturi įrašų, funkcija grąžina reikšmę *Nul.l*. Apibendrinta domeno funkcijų sintaksė: *DFunction ("lauko-varđ " arba "išraiška ", "lentelės-varđ " arba "užklausos-varđ ", "paieškos sąlyga")*

Pavyzdžiai yra pateikti KD diske: *\MSO-97\Lenteles\Chapter4.pdf TABLE 4.3:ControlSource expressions using the domain aggregate function.*

Pastaba. Domenų agregatinės funkcijos, tai specialaus formato *SQL* operatoriai, nes į funkciją galima žiūrėti kaip į užklausą, grąžinančią vieną reikšmę. Paieškos sąlyga ekvivalenti operatoriui *SQL WHERE* (nors žodis *WHERE* funkcijoje neminimas). Panaudojant funkciją visada vykdoma užklausa (operatorius *SQL*).

4.30 UŽDUOTIS.

1. Iš kompaktinio disko įkraukite atsisiųsta DB *Northwind_Ch4*, išbandykite ir patikrinkite pratimą.
2. Iš kompaktinio disko įkraukite konvertuota DB *Ch4_North-konvert*, išbandykite ir patikrinkite pratimą.
3. Peržiūrėkite savybių reikšmes.
4. Pastebėkite kitokį valdomojo elemento pavadinimą. Pakeiskite jį kitu pavadinimu.
5. Peržiūrėkite lentelę iš kompaktinio disko (*\MSO-97\Lenteles\Chapter4.pdf TABLE 4.3:ControlSource expressions using the domain aggregate function*).

4.5. Nesietinis elementas kaip kintamasis

Programavimo požiūriu kintamasis – tai atmintinėje laikinai saugoma reikšmė. Į nesietiną/laisvą formos valdomąjį elementą galima žvelgti kaip į ypatingą kintamąjį. Tokiame elemente išvedamos reikšmės egzistuoja tik tol, kol atvira forma. Nesietinis valdomasis elementas – tai globalusis kintamasis, kadangi jo reikšmė prieinama visoms duomenų bazės užklausoms, formoms ir ataskaitoms.

4.5.1. Priverstinis valdomojo elemento duomenų gavimas

Į formą galima įterpti valdomąjį elementą be reikšmės savybės *ControlSource* įvesties lauke. Tada į valdomąjį elementą tenka įvesti priverstiną reikšmę. Priverstina valdomojo elemento reikšmė gali būti įvedama klaviatūra arba *VBA* procedūra.

Pavyzdžiui, kombinuotasis paieškos blokas, kurį įjungėme į formą *Expense Reports by Employee* (žr. 1 skyr.), yra nesiejamas valdomasis elementas su savybe *ControlSource* be reikšmės. To kintamojo reikšmė yra *EmployeeID* – ieškomo pareiškėjo atpažiniklis. Kombinuotajame sąraše parinkta reikšmė „išstumia“ ją į valdomąjį elementą. Po to *VBA* procedūra sinchronizuoja formą su kombinuotojo bloko kintamojo reikšme.

4.5.2. Formos globaliųjų kintamųjų naudojimas

Nesiejamą valdomąjį elementą priklausomai nuo vartotojo ketinimų galima įterpti betarpiškai į taikomąją programą arba jungti į globaliųjų kintamųjų rinkinį. Jeigu reikia, kad reikšmės būtų prieinamos pakartotinai paleidus programą, sudarykite kintamųjų saugojimo laukelį. Tokios reikšmės tampa nuolatinėmis, nesikeičiančiomis. Bendruoju atveju formos globalieji kintamieji – tai nesiejami valdomieji elementai, kaip laikinieji kintamieji, ir siejami valdomieji elementai, kaip nuolatiniai, nesikeičiantys kintamieji. Kintamųjų reikšmės bus prieinamos kitoms formoms ir ataskaitoms, kai reikšmių forma yra atvira. Kai reikšmės įjungiamos į apskaičiavimų elementus, makrokomandas arba procedūras, *VBA* reikšmių forma gali būti uždaryta.

Kaip įterpti reikšmes į siejamus ir nesiejamus valdomuosius elementus aiškinama II ir III mokomojo leidinio dalyje.

4.6. Valdomieji elementai su dviem duomenų šaltiniais

Įprasta kad formos duomenų šaltiniu gali būti tik viena lentelė, viena užklausa arba operatorius *SQL*. Jau žinome, kad šio apribojimo galima nepaisyti, todėl galima sakyti, kad yra galimybė ieškoti duomenų reikšmių kitoje formoje arba ataskaitoje, o funkcija *DLookup()* leidžia ieškoti duomenų bet kurioje lentelėje arba užklausoje. Vadinasi, nesiejamų arba siejamų valdomųjų elementų vienines reikšmes galima gauti taikant reikšmės „ištraukimą“ (*elicit/extract*), kai panaudojamas savybės *ControlSource* įvesties laukas ir taikant reikšmės „išstūmimą“ (*pop, pull*), kai savybės *ControlSource* įvesties laukas nepildomas. Valdomųjų elementų reiškinių sąrašo blokas arba kombinuotasis blokas sudaro galimybę efektyviai peržiūrėti ir įvesti reiškinių rinkinį iš kito duomenų šaltinio.

4.6.1. Duomenų paieškos kombinuotasis blokas

Kombinuotajam ir sąrašo blokui būdinga savybė *RowSource*, kuri panaudojama valdomojo elemento ne pačios formos, bet kitam duomenų šaltiniui nurodyti. Tai reiškia, kad panaudojant savybę *RowSource* galime išvesti duomenis iš kito šaltinio: lentelės, užklaustos arba *SQL* operatoriaus minilentelės (*mini-datasheet*) duomenų eilutėmis. Be to, kombinuotasis blokas dar turi savybę *BoundColumn*. Joje galima nurodyti kuris minilentelės stulpelis pageidaujamas. Pagaliau, panaudojant savybę *ControlSource* valdomąjį elementą galima siesti su formos duomenų šaltinio lauku.

PRATIMAS. DB *Northwind_4sk* formoje *Products* kombinuotąjį bloką *SupplierID* modifikuokite išvesti papildomiems laukams iš lentelės *Suppliers*.

1. Atverkite konstravimo lange *Design* formą *Products* ir bakstelėkite kombinuotajame bloke *SupplierID*. Savybės *ControlSource* nuostata *SupplierID* rodo, kad valdomasis elementas yra siejamas su lentelės *Products* lauku *SupplierID*. Savybės *RowSource* išvesties lauke nustatytas operatorius *SQL* nuskaito eilučių *SupplierID* ir *CompanyName* reikšmes iš lentelės *Suppliers*: *SELECT DISTINCT [Suppliers].[SupplierID], [Suppliers].[CompanyName] FROM Suppliers ORDER BY [Suppliers].[CompanyName]*;

Savybei *BoundColumn* rodo, kad dabar valdomajame elemente yra įvesta pasirinktos eilutės pirmojo stulpelio reikšmė 1, o savybė *ColumnWidths* – kad atpažiniklis *SupplierID* bus slaptas (nebus rodomas), o reikšmė *CompanyName* išvedama kombinuotame sąraše. Sekančiu veiksmu papildysite sąrašą kitais stulpeliais.

Combo Box: SupplierID

Property	Value
Name	SupplierID
Control Source	SupplierID
Format	
Decimal Places	Auto
Input Mask	
Row Source Type	Table/Query
Row Source	SELECT DISTINCT [Suppliers].[SupplierID], [Suppliers].[CompanyName] FROM Suppliers ORDER BY [Suppliers].[CompanyName];
Column Count	2
Column Heads	No
Column Widths	0cm;5,794cm
Bound Column	1
List Rows	8
List Width	5,794cm
Status Bar Text	Same entry as in Suppliers table.
Limit To List	Yes
Auto Expand	Yes

(a)

SQL Statement: Query Builder

Suppliers

- * SupplierID
- CompanyName
- ContactName
- ContactTitle

Field	SupplierID	CompanyName	ContactName	Phone
Table	Suppliers	Suppliers	Suppliers	Suppliers
Sort		Ascending		
Show	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria				
or				

(b)

4.37 pav. Sąrašo lauko *SupplierID* savybės: valdomajame elemente saugomas pirmas stulpelis, o sąraše vaizduojamas antras stulpelis (a). Sąrašo šaltinio eilučių papildymas stulpeliais *ContactName* ir *Phone* (b)

2. Paleiskite veikti užklauso komponuoklį *Query Builder*: eilutės *RowSource* dešinėje bakstelėkite mygtuką *Build*. Tempkite lentelės *Suppliers* laukus *ContactName* ir *Phone* į užklauso konstravimo sritį. Uždarykite komponavimo langą, išsaugokite pakeitimus.

3. Pakeiskite kombinuotojo bloko savybes papildomiems laukams išvesti

- *ColumnCount* (stulpelių skaičius) 4

- *ColumnWidth* (stulpelių plotis coliais 1" = 2,54 cm) 0"; 1.2"; 1"; 0.5"
- *ListWidth* (sąrašo plotis) 3.2"

4. Išsaugokite formą, atvaizduokite formos *Products* vaizdą pažiūrėkite į kombinuotąjį sąrašo bloką *Supplier*. Matysite tris stulpelius iš lentelės *Suppliers* (duomenų minilentelė).

Pavlova, Ltd.	Ian Devling	(03) 444-2343
PB Knäckebröd AB	Lars Peterson	031-987 65 43
Plutzer Lebensmittelgr	Martin Bein	(069) 992755
Refrescos Americanas	Carlos Diaz	(11) 555 4640
Specialty Biscuits, Ltd.	Peter Wilson	(161) 555-4448
Svensk Sjöföda AB	Michael Björn	08-123 45 67
Tokyo Traders	Yoshi Nagase	(03) 3555-5011
Zaanse Snoepfabriek	Dirk Luchte	(12345) 1212

4.38 pav. Formos *Products* pagrindas yra lentelė *Products*, o lauko *Supplier* sąrašas išvedama minilentelė, parengta iš kitos lentelės *Suppliers*

4.6.2. Duomenys nesietinam elementui panaudojant bloko savybę *Column*

Kombinuotojo bloko arba sąrašo bloko bet kurį stulpelį galima išvesti kitame formos valdomajame elemente taikant duomenų „ištraukimą“ (pasinaudojame savybe *ControlSource*) arba „išstūmimą“ (sietino ar nesietini valdomojo elemento reikšmė nustatoma makrokomanda/procedūra *VBA*). Bet kuriuo atveju nuoroda į stulpelį ir eilutę nustatoma savybe *Column*.

PRATIMAS.

Duomenų „ištraukimo“ būdu iš lentelės *Suppliers* gaukime kontaktavimo agento pavardę *ContactName* ir telefono numerį *Phone* ir įterpkite juos į formos valdomuosius elementus taikydami savybę *ControlSource*.

1. Į formą *Products* įterpkite du nesąryšinius valdomuosius elementus – teksto blokus (žr. žemiau: Vardas ir *ControlSource*). Savybės *ControlSource* lauke įvedant išraišką sistema *Access* kiekvieną žodį apskliaučia laužtiniais skliaustais. Stulpeliai numeruojami nuo nulio. Vadinasi, *Column(0)* – tai pirmas stulpelis ir t. t.

Vardas	ControlSource
<i>txtContactName</i>	= <i>SupplierID.Column(2)</i>
<i>txtPhone</i>	= <i>SupplierID.Column(3)</i>

2. Peržiūrėjime įrašus. Nesiejamas elementas automatiškai atnaujinamas. Veikia keitimai kombinuotajame bloke. Valdomyje elemente esančių reiškinių modifikuoti negalima.

Minilentelės dviejų stulpelių duomenis galima sujungti:

1. Pašalinkite valdomuosius elementus *txtContactName* ir *txtPhone* ir įterpkite vieną nesiejamą elementą *txtContactInf* su *ControlSource* išraiška : = *SupplierID.Column(2)* & " : " & *SupplierID.Column(3)*
2. Peržiūrėkite formos keletą įrašų.

Kombinuotojo bloko stulpeliuose išvedamų duomenų tipas *Text*. Norint duomenis perduoti apskaičiavimuose reikia keisti duomenų tipą. Kai duotąsias reikšmes tipo *Text* reikia pakeisti skaičių tipo reikšmėmis, reikia pasinaudoti kuria nors viena iš 5-kių pirmųjų formulių (žr. *KD* diskas: \MSO-97\Lenteles\Chapter4.pdf TABLE4.4: *Data conversion functions*).

4.31 UŽDUOTIS.

1. Iš kompaktinio disko įkraukite konvertuota DB *Ch4_North-konvert*, susipažinkite ir patikrinkite pratimą.
2. Iš kompaktinio disko įkraukite atsisiųsta DB *Northwind_Ch4*, patikrinkite pratimo pabaigą.
3. Taikydami įgytą patirtį vykdykite pratimą su DB *Northwind_4sk*.
4. Pademonstruokite veikimą.

4.6.3. Duomenys siejamam valdomajam elementui

Siejamą lauko savybę *ControlSource* rodo tik siejamą lauką ir netinka reikšmei gauti. Jeigu reikia į siejamą valdomąjį elementą įvesti reikšmę iš kombinuotojo bloko arba sąrašo bloko minilentelės, reikia parengti makrokomandą arba procedūrą *VBA*, taikyti duomenų „išstūmimo“ būdą.

Pavyzdžiui, DB *Northwind* formos *Orders* kombinuotojo bloko reikšmei *CustomerName* įvesti į teksto bloką *ShipName* ir adresui įvesti į valdomuosius elementus, rodančius tiekimo adresus (*Bill To*) yra panaudojama procedūra *VBA*. Apie tai bus išsamiai aiškinama mokomosios knygos II ir III dalyje.

4.7. Formų tarpusavio sąveika

Panagrinėsime dviejų atvirų formų galimus tarpusavio saitus. Jeigu atvertos dvi formos su skirtingais duomenų šaltiniais, sąlygos tarp tų formų tos pačios, kokios buvo duomenų šaltinių – tarp lentelių bei užklausų.

4.7.1. Dviejų formų sinchronizavimas

Tarpusavyje siejamų dviejų formų sinchronizavimas yra toks suderinimo atvejis, kai abiejose formose išvedamos atitinkamos reikšmės. Sinchronizavimo procesą gali valdyti bet kuri forma. Pagal vienos formos užsakymą antroje formoje vyksta duomenų filtravimas.

Formų sinchronizavimą geriausiai užtikrina programavimas. Be to, galima panaudoti sistemos taikomąją programą *Form Wizard*.

Sinchronizavimas taikant programą *Form Wizard*. Taikomąją programą *Form Wizard* galima parengti dvi formas, siejamas duomenų šaltiniais, be to, sukurti formų sinchronizavimo procedūras.

PRATIMAS. PANAGRINĖKITE, KAIP VEIKIA TAIKOMOJI PROGRAMA – MEISTRAS FORM WIZARD.

Reikia sinchronizuoti formas, parengtas pagal lenteles *Customers* ir *Orders*.

1. DB *Northwind-sk4:Database* lange, atverkite kortelę *Forms*, toliau: *New*, nurodykite *Forms Wizard* ir *OK*.
2. Pirmajame *Form Wizard* lange pasirinkite lentelį *Customers* ir *Orders* keletą laukų, kurie bus įjungti į formas.
3. Kitame lange reikia nurodyti, kuri forma valdys sinchronizavimą. Kai pažymėta *by Customers* įjunkite jungiklį parinkties *Linked forms*, susietos formos, ir bakstelėkite *Next*.
4. Parinkite formos stilių, pavyzdžiui, *Standard*.
5. Pavadinkite formas: *CustomersLinked* ir *OrdersLinked*. Bakstelėkite *Finish*.
6. Taikomoji programa *Form Wizard* parengia formas ir išveda sinchronizavimo procesą valdančią formą *CustomersLinked*.
7. Kai bakstelėsite mygtuką – perjungiklį *OrdersLinked* ekrane matysime formą *OrdersLinked*, kurios įrašai dabar parinkti veikiant filtrui.
8. Formoje *CustomersLinked* nustatykite kitą įrašą ir peržiūrėkite formoje *OrdersLinked* filtro automatiškai išvestus įrašus.

Kol forma *OrdersLinked* atvira, mygtukas-perjungiklis formoje *CustomersLinked* išlieka nuspauštas. Kai bakstelėsite nuspauštą mygtuką, forma *OrdersLinked* uždaroma ir mygtukas užima pradinę padėtį.

4.39 pav. Pagalbine programa *Form Wizard* siejant dvi formas *Customers* ir *Orders* automatiškai įrengiamas mygtukas *OrdersLinked* formų įrašams sinchronizuoti

Programai, kurią parengė *Form Wizard*, paleisti pasitarnauja du įvykiai:

- Įvykis *Click*, kai bakstelimas mygtukas-perjungiklis. Jeigu forma *OrdersLinked* atverta, tai bakstelėdami mygtuką ją uždarote, jeigu uždarytą – bakstelėdami ją atveriate, kartu parengiamas filtras įrašams išrinkti.
- Įvykis *Current* formai *CustomersLinked*. Peržiūrint eilinio pareiškėjo įrašą forma atpažįsta įvykį *Current*. Jei tuo metu forma *OrdersLinked* uždaryta, jokio veiksmo nėra, o jeigu atverta sudaromas formos *OrdersLinked* įrašams filtruoti tinkamas filtras.

Procedūros VBA. Programa *Form Wizard* įvykiams *Click* ir *Current* apdoroti parengia dvi procedūras: *Sub ToggleLink_Click()* ir *Sub Form Current()*. Be to, programa *Form Wizard* parengia keturias papildomas (*private*) procedūras, kurios papildo pirmąsias dvi. Tokios procedūros vadinamos paramos procedūromis (*procedure support*). Visos 4 procedūros saugomos formos modulio srityje *General* (langas- *CustomersLinked Class Module*).

Dirbant su formomis *CustomersLinked* ir *OrdersLinked* kartu atskirai (formas galima įkelti taip, kaip įprasta ir iš duomenų bazės lango) galima suprasti, kad egzistuoja ir kiti – papildomieji įvykiai.

Nepriklausomai nuo formos iškėlimo būdo, mygtukas-perjungklis išlieka nuspaustas, kai forma atverta ir pradinėje padėtyje (nenuspaustas), kai forma uždara. Mygtuko-perjungklio elgsena diktuoja *Form Wizard* vykdyti programą atsižvelgiant į įvykius, kurie lydi jo būsenos kitimą. Tie du įvykiai *Load* ir *Unload* yra atpažįstami formos *OrdersLinked*.

4.32 UŽDUOTIS.

Peržiūrėkite procedūras, kurias galėsite pamatyti, kai formos (vienos ir kitos) konstravimo lange bakstelsite mygtukų juostoje mygtuką *Code*. Čia žemiau pateikiami kai kurių procedūrų kodai.

```
Sub Form_Current()  
On Error GoTo Form_Current_Err  
If ChildFormIsOpen() Then FilterChildForm  
Form_Current_Exit:  
Exit Sub  
Form_Current_Err:  
MsgBox Error$  
Resume Form_Current_Exit  
End Sub  
  
Sub ToggleLink_Click()  
On Error GoTo ToggleLink_Click_Err  
If ChildFormIsOpen() Then  
CloseChildForm  
Else  
OpenChildForm  
FilterChildForm  
End If  
ToggleLink_Click_Exit:  
Exit Sub  
ToggleLink_Click_Err:  
MsgBox Error$  
Resume ToggleLink_Click_Exit  
End Sub  
  
Sub Form_Load()  
On Error GoTo Form_Load_Err  
If ParentFormIsOpen() Then Forms![CustomersLinked]!ToggleLink = True  
Form_Load_Exit:  
Exit Sub  
Form_Load_Err:  
MsgBox Error$  
Resume Form_Load_Exit  
End Sub  
  
Sub Form_Unload(Cancel As Integer)  
On Error GoTo Form_Unload_Err  
If ParentFormIsOpen() Then Forms![CustomersLinked]!ToggleLink = False  
Form_Unload_Exit:  
Exit Sub  
Form_Unload_Err:  
MsgBox Error$  
Resume Form_Unload_Exit  
End Sub
```

4.7.2. Sinchronizavimas pagrindinės ir pagalbinės formos

Sistema *Access* pagrindinę formą su pagalbine forma sinchronizuoja automatiškai. Čia tik apžvelkime, kaip veikia toks sinchronizavimas. (Panašiai sinchronizuojama ataskaita su pagalbine ataskaita arba ataskaita ir forma)

Valdomojo elemento „pagalbinė forma“ savybės yra trys: *SourceObject*, *LinkChildFields* ir *LinkMasterFields*. Kai pagalbinės formos savybei *SourceObject* suteikiamos pagrindinės formos vardas, pagalbinė forma siejama su pagrindine. Jeigu kitos dvi savybės nenurodomos (išlieka tuščios), tai formų saitas išlieka tačiau įrašus galima peržiūrėti atskirai.

Jeigu dviejų formų įrašų šaltiniai yra susieti, tai galima ir formose įrašus susieti. Kai saito laukai bent keli, savybės lauke jie skiriami dvitaškiu.

Pagalbinė programa *Subform/Subreport Linker* sudaro galimybę suteikti įrašų saitų savybes. Programą paleisite atvėrę pagrindinę formą konstravimo (*Design*) lange. Išveskite pagalbinės formos savybių sąrašą ir bakstelėkite mygtuką *Build* bet kurioje saito savybių eilutėje.

Dialogo lange *Subform Field Linker* nurodomi formų saito laukai.

Realizuojant metodą „forma/pagalbinė forma“ vartotojas tik nustato valdomojo elemento „pagalbinė forma“ savybes, o sistema *Access* pati sinchronizuoja formas. Esant tam tikromis sąlygomis sistema pati nustato charakteristikas.

Jeigu pagalbinė forma arba pagalbinė ataskaita parengiama iš duomenų bazės lango tempimo būdu į kitą formą/ataskaitą, sistema *Access* automatiškai išveda sudėtinį valdomąjį elementą ir nustato įrašų saito savybes, tačiau tik tada, jei galioja dvi sąlygos:

- o pagrindinės formos arba ataskaitos šaltinis yra lentelė su pagrindiniu raktu, nes sistema *Access* suteikia savybę *LinkMasterFields* pirminio rakto laukui;
- o pagalbinės formos arba pagalbinės ataskaitos šaltinis yra užklausa arba lentelė su lauku (arba laukais), kurio vardas toks pat kaip pagrindinės formos lentelės pagrindinio rakto (arba suderinamų dokumentų tipo); *Access* suteikia savybei *LinkChildFields* pagalbinės formos arba pagalbinės ataskaitos duomenų šaltinių atitinkamai tapačius vardus.

Metodas „forma/pagalbinė forma“ gali būti taikomas, kai lentelės siejamos sąlyga „vienas (pagrindinė forma) su daugeliu (pagalbinė forma)“. Tokia sąlyga siejamos lentelės *Customers* ir *Orders*. Pagrindinės formos lentelė *Customers*, pagalbinė - *Orders*.

4.33 UŽDUOTIS.

1. Įkraukite konvertuotą DB *Ch4_North-konvert*.
2. Iškelkite/eksportuokite į DB *Northwind_4sk* sudėtinę formą *CustomerswithOrders*, parengtą iš lentelių *Customers* ir *Orders* ir pagalbinę formą *CustomerswithOrders Subform* (*File* → *Export*, nurodysite kur, nurodysite ką).
2. Peržiūrėkite sudėtinę formą ir pamatykite pagrindinės bei pagalbinės dalies savybes.
3. Sukonstruokite savo panašią sudėtinę formą (formą su pagalbine forma) *Customers with Orders* panaudodami paslaugų programą *Form Wizard*.
4. Peržiūrėkite formas, savybes, procedūras.

4.7.3. DVIEJŲ PAGALBINIŲ FORMŲ SINCHRONIZAVIMAS

Programa *Form Wizard* gali sinchronizuoti pagrindinę formą su pirma pagalbine forma ir antrą pagalbinę formą su pirma pagalbine forma.

Duomenų bazės *Northwind* forma *Customer Orders* demonstruoja tokią *Form Wizard* galimybę. Formoje dvi sinchronizuotos pagalbinės formos. Antra pagalbinė forma pateikia paraiškos detales, jos šaltinis - užklausa *Order Details Extended*.

PRATIMAS.

Patyrinėkite, kaip taikomoji programa *Form Wizard* automatizuoja sinchronizavimą:

1. Atverskite konstravimo lange formą *Customer Orders*. Pagrindinė forma valdo pirmos pagalbinės formos sinchronizavimą. Pirmos pagalbinės formos savybių sąrašas rodo, kad lentelės šaltinio įrašų saitas pagal lauką *CustomerID* tarp pagrindinės formos ir pagalbinės formos *Customer Orders Subform1*. Antros pagalbinės formos savybės rodo, kad saitas tarp pagalbinių formų nustatomas pagal lauką *OrderID*.
2. Atverkite konstravimo lange pirmą pagalbinę formą *Customer Orders Subform1*. Formos savybių sąrašas rodo, kad *Form Wizard* parengė įvykio apdorojimo procedūrą, pradedančią veikti pagal formos įvykį *Current*. Forma atpažįsta įvykį *Current* po įrašo aktyvinimo ir prieš perėjimą į kitą įrašą. Be to, - pirmą kartą atveriant formą ir prieš išvedant pirmą įrašą nustatomas įvykis *Current*.
3. Peržiūrėkite *VBA* procedūrą *Sub Form_Current()*. Įvykio procedūrą *Event Procedure* išvesite, kai bakstelėsite mygtuką *Build* savybės *OnCurrent* eilutės dešinėje. Turinį nagrinėsime vėliau. Dabar atkreipkite dėmesį į eilutę, kurios pabaigoje yra žodis *Requery*, - tai antros papildomosios formos užklausa.

Ši procedūra vykdoma ir iškviečia antrą pagalbinę formą, kai pirmoje pagalbinėje formoje bakstelima ties kitu įrašu.

4.8. Santrauka

Daugelį *Access* automatizuotų taikomųjų programų sieja vien tik formos ir ataskaitos. Duomenų bazės automatizuojamos parengiant ir taikant makrokomandas ir *VBA* procedūras.

- Formos savybė *RecordSource* rodo ar forma arba ataskaita yra siejama su lentele arba užklausa. Savybės įvesties lauke rašomas lentelės, užklauskos arba operatoriaus *SQL* vardas. Taip sietinė forma arba ataskaita yra siejama su savo duomenų šaltiniu. Jeigu savybė *RecordSource* yra neužpildyta (tuščia), tai forma arba ataskaita yra nesietinė.
- Jeigu formai, kaip antras duomenų šaltinis yra taikoma priemonė (užklausa) *AutoLookup* ir yra suformuojamas duomenų šaltinis iš dviejų lentelių, kurių saito sąlyga „vienas su daugeliu“, tada, kai tik įvedama bendrojo lauko reikšmė, galima automatiška duomenų paieška kitoje lentelėje.
- Įprasta, kad forma arba ataskaita turi vieną duomenų šaltinį, tačiau yra keletas būdų informacijai gauti iš kitų šaltinių:

Apskaičiavimo valdomasis elementas. Duomenų paieškai lentelėje arba užklausoje savybės lauke *ControlSource* galima panaudoti funkciją *Lookup*. Išraiška leidžia gauti duomenis ir juos įkelti į valdomąjį elementą (ištraukimo būdas).

Programavimas. Galima rasti arba apskaičiuoti reikšmę, o po to ją įkelti į nesiejamą valdomąjį elementą (išstūmimo būdas), kurio savybė *ControlSource* nenurodyta, arba į siejamą valdomąjį elementą.

Kombinuotojo bloko arba sąrašo valdomasis elementas. Tokius valdomuosius elementus galima sieti su dviem skirtingom lentelėm arba užklausom. Tokius valdomuosius elementus galima sieti su formos duomenų tipo lauku ir išvesti įrašus iš kito duomenų šaltinio. Patogu perteikti reikšmes formai iš kitos lentelės arba užklauskos.

- Jeigu formoje einamieji duomenys neatnaujinami natūraliu būdu, einamųjų duomenų išvedimą galima automatizuoti taikant programavimą.
- Kai konstruojama sudėtinė forma pagal siejamas lenteles ar užklausas, valdomuosius elementus galima sinchronizuoti pagal siejamų formų savybes (pagrindinė forma/pagalbinė forma).
- Atskiruose languose išvedamų siejamų formų sinchronizavimas galimas tik taikant programavimą.

Ketvirtuoju skyriumi baigiama duomenų bazių automatizavimo įvadinė dalis ir trumpa makrokomandų bei *VBA* programavimo panaudojimo apžvalga. Dabar galima žengti toliau. Galima mokytis rengti makrokomandas be programavimo - nagrinėti makroprogramavimą (žr.: II knyga, 5 - 9 skyriai). Turintys daugiau patirties gali praleisti makrokomandų rengimą ir iš karto mokytis programuoti - nagrinėti programavimą *Access VBA* (žr.: III knyga, 10 - 20 skyriai).