

Pirmoji pažintis su Delphi

Jonas Blonskis, Vytautas Bukšnaitis, Valentina
Dagienė, Vacius Jusas, Romas Marcinkevičius

2003 metų pavasarį pasirodys programavimo pagrindų Delphi vadovėlis XI–XII klasėms, kurį parašė šio straipsnio autoriai. Straipsnyje apžvelgiami svarbiausieji programų kūrimo Delphi terpėje principai.

Delphi — tai objektiškai orientuotų programų kūrimo sistema, kurios priemonėmis galima parengti įvairaus lygio ir paskirties programas: tiek mėgėjiškas, tiek profesionalias. Programos *Delphi* terpėje kuriamos, įtraukiant į programą reikalingus objektus (komponentus), suteikiant jiems reikalingas savybes ir programuojant *Object Pascal* kalba reakcijas į įvykius, kurie gali įvykti programos vykdymo metu. Visas tas dalis į vientisą *Object Pascal* kalbos sintaksę atitinkančią programą automatiškai jungia *Delphi* sistema. Kadangi *Delphi* siūlo daug paruoštų įvairios paskirties komponentų, kuriuos programų kūrėjai gali įtraukti į savo programas, programuotojams labai sumažėja juodo darbo.

Pirmoji *Delphi* sistemos versija *Windows* operacijų sistemai pristatyta 1995 metais. Vėliau (2001 m.) sukurta *Kylix — Delphi* versija *Linux* operacinei sistemai. Nuo 2002 m. pabaigos galima įsigyti *Delphi 7* versiją. Kiekviena aukštesnė versija suteikia naujų galimybių duomenų bazių valdymo sistemoms, interneto ir kt. programoms kurti.



Norint išmokti kurti sudėtingas programas, reikia ne tik skaityti svetimas programas, bet ir pačiam programuoti. Todėl skaitytojui patariame įsigyti kurią nors *Delphi* versiją, įdiegti ją savo kompiuteryje ir tik po to tęsti *Delphi* studijas.



Delphi terpė

Pradėjus dirbti *Delphi* (sakysime, kad sistema jau įdiegta kompiuteryje), atveriamą integruota programų kūrimo terpė, kurioje yra visos priemonės, reikalingos vartotojo programoms projektuoti, kurti, testuoti ir derinti. Ekране šią terpę sudaro 4 atskiri langai: pagrindinis *Delphi* langas (antraštė: *Delphi3 — Project1*), objektų inspektoriaus langas (*Object Inspector*), programos pagrindinės formos grafinio vaizdo langas (*Form1*) ir pagrindinės formos modulio programos langas (*Unit1.pas*), kuris gali būti paslėptas po *Form1* langu (1 pav.).

Kiekvieną iš šių langų galima nepriklausomai nuo kitų valdyti: didinti, mažinti, perkelti į kitą ekrano vietą, užverti, atverti. Užvėrus pagrindinį langą, *Delphi* sistema baigia darbą.

Komandų meniu — tai visų *Delphi* darbo metu galimų veiksmų rinkiniai. Kai kurie komandų rinkiniai, pavyzdžiui, *File*, *Edit*, *Help*, yra panašūs į kitų taikomųjų programų analogiškus rinkinius, o kai kurie, pavyzdžiui, *Project*, *Run*, būdingi tik programų kūrimo terpėms. Komandų paskirtis paaiškės programų kūrimo metu.

Dažniausiai naudojamos komandos pakartotos ir mygtukų juostose. Jų pavadinimai ir atliekami veiksmai sutampa su atitinkamų komandų pavadinimais ir atliekamais veiksmais. Dažniausiai mums prireiks šių mygtukų:  — saugoti viską (*Save all*),  — vykdyti programą (*Run*),

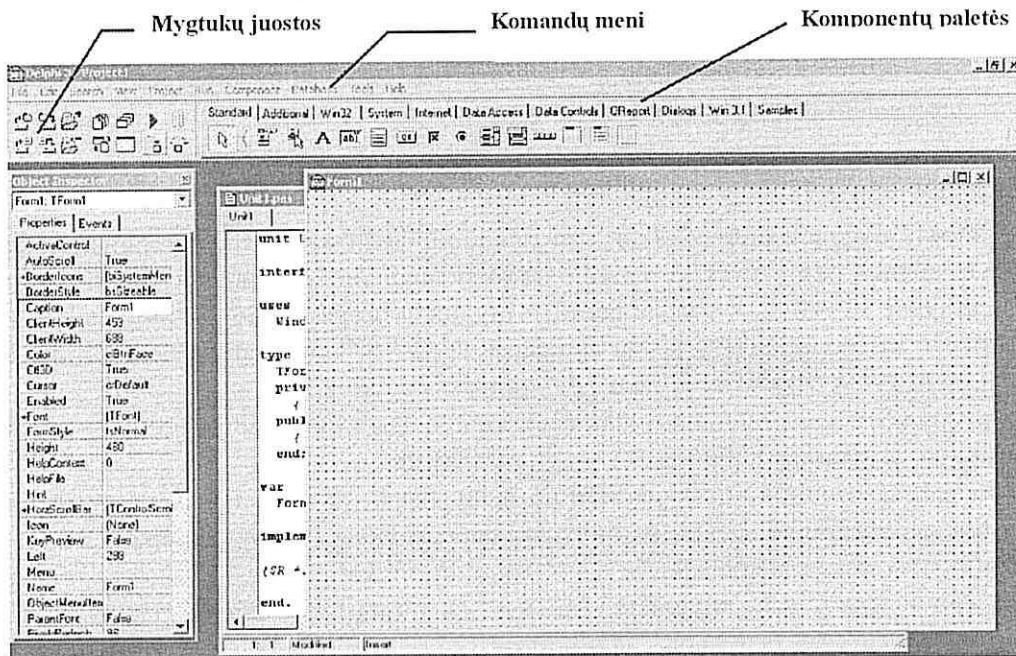
 — atverti projektą (Open Project),  — iš formos lango pereiti į modulio langą ir atvirkščiai (Toggle Form/Unit).

Komponentai, jų savybės ir metodai, įvykiai

Komponentai — tai objektai, iš kurių sudaroma *Delphi* terpėje kuriama programa. Visi komponentai yra griežtai tipizuoti ir vartotojo programose naudojami pagal to tipo komponentų paskirtį. Pavyzdžiui, programos mygtukams kurti gali būti taikomi `TButton` tipo komponentai. Pagrindiniame *Delphi* lange komponentai yra sugrupuoti pagal paskirtį į atskirus rinkinius (paletes): `Standard`, `Additional`, `Win32` ir t.t.

Vartotojo programoje visada bus bent vienas komponentas — forma (`TForm1` tipo programos langas). Šis komponentas įkeliamas į kuriamą programą automatiškai, tik pradėjus *Delphi* darbą. Visi kiti reikalingi komponentai įtraukiami į programą tokiu būdu: reikia spragtelėti pele reikalingo komponento mygtuką komponentų paletėse, o po to — spragtelėti programos formos lange. Komponento matmenis ir vietą formos lange taip pat galima keisti pele. Tokiu pačiu būdu galima keisti ir programos pagrindinės formos lango matmenis bei vietą ekrane. Spragtelėjus ties komponentu dešiniuoju pelės klavišu, galima iškviešti kontekstinį meniu, kuriame `Align...` komanda galima nustatyti tikslią komponento vietą (pavyzdžiui, lango centre), o `Size...` komanda — komponento matmenis. Į programą galima įtraukti bet kokį skaičių tiek to paties, tiek skirtingų tipų komponentų. Visiems jiems *Delphi* suteikia unikalius vardus, atskleidžiančius jų tipą. Pavyzdžiui, `TLabel` (etiketė) tipo komponentams, kurie naudojami įvairiems užrašams programos lange, suteikiami vardai `Label1`, `Label2` ir t.t.

Kiekvienas programos komponentas, kaip ir realaus pasaulio objektai, turi tam tikras savybes. Vienos jų savybės, pavyzdžiui, antraštė, spalva ir pan., būdingos beveik visų tipų komponentams, kitos — tik vieno kurio nors tipo komponentams. Dažniausiai keičiamas kuriamos programos komponentų savybes galima peržiūrėti ir keisti objektų inspektoriaus (Object Inspector) lange, `Properties` plokštelėje (1 pav.).



1 pav. Pagrindinis *Delphi* langas



Šiame lange rodomos savybės to programos objekto, kuris tuo metu yra aktyvus (prieš tai jame buvo spragtelėta pele). Kairiajame stulpelyje abėcėlės tvarka surašyti savybių pavadinimai, dešiniajame — reikšmės. Pakeitus kurios nors savybės reikšmę, atitinkamai pakinta objekto vaizdas programos lange (aišku, jei ši savybė nusako objekto vaizdą). Pavyzdžiui, pakeitus `Label1` komponento `Caption` (antraštė) savybės reikšmę, pakinta užrašas objekto langelyje programos formos lange.

Su kiekvienu programos objektu galima atlikti tam tikrus veiksmus (vykdyti objekto metodus). Galimų metodų rinkinys taip pat priklauso nuo objekto tipo. Objektų metodai dažniausiai taikomi programuojant veiksmus, kuriuos programa turėtų atlikti reaguodama į vienus ar kitus įvykius programos vykdymo metu. Įvykių, į kuriuos gali reaguoti objektas, sąrašas taip pat matomas objektų inspektorius lange, `Events` plokštelėje (1 pav.). Objekto reakcijos į vieną ar kitą įvykį (pavyzdžiui, mygtuko spragtelėjimo pele — `OnClick`) veiksmus turi suprogramuoti programos kūrėjas.

Programos objektų savybių keitimas


Tiek programos pagrindinės formos, tiek kitų į programą įtrauktų komponentų savybes galima apibrėžti (pakeisti) dviem būdais: 1) programos kūrimo ar modifikavimo metu, 2) programos vykdymo metu.

Kurimo metu objektų savybių reikšmės nurodomos objektų inspektorius lange. Savybės reikšmė, gali būti:

- simbolių seka arba skaičius: naują reikšmę galima tiesiog užrašyti atitinkamoje objektų inspektorius lango eilutėje;
- viena iš kelių galimų reikšmių: naują reikšmę galima išrinkti iš sąrašo paspaudus sąrašo išskleidimo mygtuką  tos savybės eilutės dešinėje;
- parametrų rinkinys: naujas savybę nusakančių parametrų reikšmės galima nustatyti dialogo lange, kuris pasirodo paspaudus dialogo lango atvėrimo mygtuką  savybės eilutės dešinėje.

Programos vykdymo metu savybių keitimo veiksmai turi būti rašomi programoje *Object Pascal* kalba. Dažniausiai tam naudojamas priskyrimo sakiny.

Vartotojo programos kūrimas, saugojimas ir vykdymas

Darbo pradžioje į programą automatiškai įtraukiamas vienas `TForm1` tipo komponentas — pagrindinės formos langas `Form1`. Nors programa dar tik pradėta kurti, iš esmės ji turi visas programos savybes ir ją galima vykdyti — paspaudus `Run` mygtuką, programa kompiliuojama ir vykdoma. Ekrane pasirodo tuščias pilkas langas su antrašte `Form1`. Šį langą galima pele kilnoti, keisti jo dydį. Lango uždarymo mygtukas  nutraukia programos darbą.

Programos naujos dalys įtraukiamos kaip nauji komponentai į pagrindinę programos formą arba kaip papildomos programos kodo dalys į programos tekstą.

Programa galima naudotis ne tik jos kūrimo metu, bet ir vėliau, jei ji išsaugota kompiuterio diske. Kiekvienai programai saugoti rekomenduotina pasidaryti atskirą katalogą, kadangi programos kompiliavimo metu sistema sukuria iki 10 bylų. Po to, kai padarysite katalogą, viską galima išsaugoti vienu — `Save All` mygtuku. Atliekant tai pirmą kartą, dialogo languose reikia nurodyti pasirinktą katalogą, programos pagrindinės formos modulio bei programos projekto bylų vardus. Sistema siūlo `Unit1.pas` ir `Project1.dpr` vardus, kurių galima nekeisti. Kitoms byloms vardus suteikia pati sistema. Norint neprarasti to, kas padaryta programos kūrimo metu,

Save All mygtuką rekomenduotina paspausti po kiekvieno programos keitimo. Tai užima labai nedaug laiko, nes bylų vardus reikia nurodyti tik pirmo saugojimo pradžioje.

Anksčiau sukurta programa įkeliama į *Delphi* aplinką `File.Open Project` komanda arba `Open Project` mygtuku.

Programos struktūra

Delphi terpėje kuriama programą sudaro viena pagrindinė dalis — programos projektas (*project*) ir viena arba daugiau papildomų dalių — modulių (*unit*).

Programos projektas saugomas tokiose bylose:

- .dpr (pavyzdžiui, `Project1.dpr`) — projekto pagrindinė byla (tekstinė),
- .res (`Project1.dpr`) — projekto resursų (piktogramų ir pan.) byla (ne tekstinė),
- .dof (`Project1.dof`) — projekto parametrų byla (tekstinė),
- .exe (`Project1.exe`) — projekto vykdomoji byla (ne tekstinė).

Programos moduliai saugomi tokiose bylose:

- .pas (pavyzdžiui, `Unit1.pas`) — formos `TForm1` programos kodo byla (tekstinė),
- .dfm (`Unit1.dfm`) — formos vaizdo byla (ne tekstinė, tačiau *Delphi* terpėje galima matyti kaip tekstą),
- .dcu (`Unit1.dcu`) — modulio byla po kompiliavimo (ne tekstinė).

Programos projekto bylą `Project1.dpr` galima pamatyti `View.Project Source` komanda pagrindiniame *Delphi* lange.

Saugant programą po kokių nors pakeitimų, diske paliekamos ir ankstesnės bylų versijos, pakeitus jų vardų plėtinius. Pavyzdžiui, `Unit1.pas` senesnioji versija yra `Unit1.pa`.

Dalį bylų *Delphi* sistema automatiškai generuoja iš grafinio vaizdo, kurį sukuria programuotojas, o kitą dalį — programos kompiliavimo metu. Kaip minėta, programuotojui *Object Pascal* kalba reikia suprogramuoti tik reakcijos į įvairius įvykius veiksmus. Šie veiksmai rašomi `.pas` bylose.

Nerekomenduotina pačiam taisyti projekto bylos. Pradedančiajam programuotojui jo tekstas naudingas tuo atveju, kai norima pasitikrinti, kur saugomi programos moduliai.

Programos kūrimo metu į formos langą įkeliama vis nauji komponentai (mygtukai — `Button`, etiketės — `Label` ir pan.). Tuo metu sistema automatiškai papildo `Unit1.pas` byloje esanti `TForm1` klasės aprašą.

Sudėtingose programose gali būti ne viena, bet kelios formos. Kiekvienai iš jų kuriama atskira klasė (`TForm2`, `TForm3`, ...) ir sukuriama atitinkamos bylos. Be to, programuotojas savo reikmėms gali kurti ir kitos paskirties klases, kurios paprastai saugomos atskirose bylose.

Visiškai suderinta programa gali būti vykdoma nesinaudojant *Delphi* terpe. Tam yra naudojama programos vykdomoji byla (`Project1.exe`).

`TForm1` klasės struktūra

Kaip minėta, *Delphi* terpėje kuriama programa susideda iš komponentų (objektų), turinčių tam tikras savybes ir metodus. Bendru atveju objektą sudaro laukų, savybių ir metodų rinkinys. *Object Pascal* kalboje objekto tipui nusakyti naudojama *klasės* sąvoka. Klasės laukai yra tai, ką klasė žino, metodai — tai, ką klasė moka. Savybės nusako keičiamas objekto charakteristikas.

Programų pradžioje visada pateikiamas klasės aprašas, o tik po to — tos klasės objektų aprašai. Programoje galima naudoti bet kokį skaičių tos pačios klasės objektų. Klasių (objektų tipų) skaičius taip pat nėra ribojamas.

Klasės yra objektiškai orientuotų programų pagrindas. Plačiau apie jų kūrimą ir naudojimą bus kalbama vėliau. Pirmosiose programose pakaks išmokti papildyti pagrindinio programos objekto — `Form1` komponento klasę.

TForm1 aprašas type dalyje pradedamas eilute:

```
TForm1 = class (TForm)
```

Toliau surašomi į formą įtrauktų kitų komponentų vardai (kartu nurodoma ir komponento klasė) ir metodų antraštės. Po paskutinio užrašomas tarnybinis žodis end. Tai padaro pati sistema formos kūrimo metu. Metodų tekstai surašomi toje pačioje byloje po žodžio implementation. Kiekvienas metodas pradedamas antrašte, o tarp tarnybinių žodžių begin ir end programuotojas turi nurodyti metodo viduje atliekamus veiksmus. Atkreipkite dėmesį į tai, kad klasių vardai pradedami T raide.

TForm1 klasės apraše rasite žodžius private ir public. Tiek po žodžio private, tiek po žodžio public programuotojas gali rašyti savo kintamuosius ir metodų antraštes.

Form1 komponento aprašą var dalyje sudaro viena eilutė:

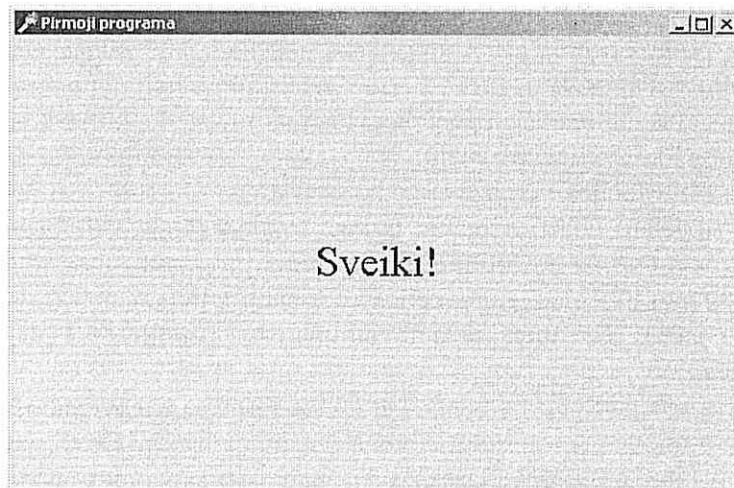
```
Form1:TForm1;
```

Taip pat (vardas:tipas) aprašomi visi *Delphi* programų kintamieji.

Pirmoji programa

Užduotis. Sukurkite programą, kuri atskirame lange parodytų užrašą „Sveiki!“.

Tokia programa paprastai pradedama pažintis su bet kuria programavimo kalba. Programos langas galėtų atrodyti taip, kaip parodyta 2 paveiksle.



2 pav.

Aprašysime detaliai, kaip ši programa kuriama, išvardysime visus atliekamus veiksmus.



Pirmasis žingsnis. Pasiruošimas darbui.

Sukuriamas katalogas, kuriame bus saugomi visos šios programos bylos. Katalogą galima pavadinti vardu PirmojiPrograma.

Paleidžiama dirbti *Delphi*. Ekrane *Delphi* terpėje sukuriamas ir naujos programos pagrindinis — pilkas Form1 langas. Tai tipiškas Windows aplinkos langas. Forma tuščia. Po ja yra programos teksto langas Unit1.pas. Čia jau yra programos dalis, kuri aprašo sukurtą formą.

Mygtuku Save All programa rašoma į diską. Ekrane pasirodo Save Unit1 As dialogo langas. Nustatomas PirmojiPrograma katalogas. Paliekamas siūlomas formos programos bylos vardas Unit1.pas ir spaudžiamas mygtukas Save. Vietoje šio dialogo lango pasirodo


kitas — Save Project1 As langas. Jame taip pat paliekamas siūlomas projekto bylos vardas Project1.dpr. Paspaudus Save, sukuriamos visos reikalingos programos bylos.


Nors nereikėjo nieko programuoti, jau yra programa, kurią galima kompiliuoti ir vykdyti Run.Run komanda arba  mygtuku. Ekране gaunamas tuščias programos langas, turintis visas Windows terpės langų savybes: langas turi antraštę ('Form1'), galima keisti jo vietą ekrane bei matmenis ir uždaryti (baigti programą)  mygtuku.

Antrasis žingsnis. Formos lango savybių keitimas.

Esant aktyviam Form1 langui, Object Inspector lange Properties plokštelėje yra Form1 savybių sąrašas: savybės vardas ir reikšmė. Savybių vardai surašyti abėcėlės tvarka. Vienos iš savybių — antraštės (Caption) reikšmė Form1 pakeičiama PirmojiPrograma. Iš karto pasikeičia ir Form1 lango viršuje matomas užrašas. Panašiai galima keisti ir kitas savybes, tačiau, kaip minėta, nepatartina keisti komponento vardo (Name savybė). Išmokite pakeisti Form1 lango matmenis (ClientHeight ir ClientWidth) ir spalvą (Color).

Trečiasis žingsnis. Komponento, reikalingo tekstui užrašyti, kūrimas ir pildymas.

Standard komponentų paletėje pele paspaudžiamas  (Label) mygtukas. Taip išrenkamas TLabel klasės komponentas, kurį pelės spragtelėjimu galima padėti į bet kurią Form1 lango vietą. Sukuriamas stačiakampis laukas su jo viduje esančiu pavadinimu Label1. Kartu į Form1 įtraukiamas naujas objektas — Label1 komponentas. Jei šis komponentas yra aktyvus, Object Inspector lange matomas jo savybių sąrašas. Pakeiskime Caption reikšmę iš Label1 į Sveiki!. Paspaudus dešinį pelės mygtuką, iškviečiamas kontekstinis meniu¹. Label1 komponentas padedamas tiksliai į formos centrą (Align... komandoje pasirenkamas Horizontal ir Vertical išdėstymo variantas Center in Window).

Label1 komponento Font savybės lauke spragtelėjus pele, pasirodo laukelis su mygtuku . Paspaudus šį mygtuką, atsiveria Font dialogo langas su paslaugų sąrašais. Nurodome tokias reikšmes:

Font:Times New Roman, Font Style:Regular, Size:24, Color:Black, Script:Baltic.


Paspaudus mygtuką OK, Label1 komponento langelyje užrašomas tekstas nurodyto tipo, stiliaus ir dydžio šriftu.

Ketvirtasis žingsnis. Programos bandymas.

Save All mygtuku pagrindiniame Delphi lange išsaugomi visi padaryti veiksmai.

Pagrindinio sistemos lango Run mygtuku kompiliuojama ir vykdoma programa. Jeigu kompiliuojant programą yra klaidų, programos teksto lango (Unit1.pas langas) apačioje pasirodo pranešimai. Jeigu klaidų nėra, programa iš karto pradeda vykdyti.

Ekране gaunamas langas vardu Pirmoji Programa. Lango centre bus tekstas Sveiki!. Kartu pasikeičia ir Delphi pagrindinio lango antraštė: atsiranda papildomas užrašas [Running].

Mūsų programos langas turi visas Windows langų savybes: galima pele perkelti į kitą ekrano vietą, keisti dydį, dešinėje viršuje jis turi tris mygtukus. Sumažinus lango dydį tiek, kad jame nebetilptų užrašas, kraštuose atsiranda peržiūros juostos. Programos darbo langas uždaromas (tuo pačiu baigiamas programos darbas) mygtuku . Kitas — avarinis — programos darbo nutraukimo būdas: vykdyti pagrindinio Delphi meniu komandą Run.Program Reset.

¹ Kontekstinis meniu priklauso nuo situacijos (konteksto), kurioje jis iškviečiamas.

TForm1 klasės metodų kūrimas

Tiek Form1, tiek jos viduje esantys objektai gali reaguoti į tam tikrus įvykius. Jų sąrašas pasirodo *Delphi Object Inspector* lango *Events* plokštelėje. Norint užrašyti reakcijos į įvykį metodą, reikia atlikti tokius veiksmus:

pele išrenkame reikalingą objektą, pavyzdžiui, formą Form1,

Events plokštelėje surandame reikalingo įvykio pavadinimą ir jo eilutėje dešinėje pusėje du kartus spragtelime pele.

Delphi sistema užrašys *Events* plokštelėje metodo vardą, *Unit1.pas* byloje TForm1 klasės apraše užrašys metodo antraštę, o *implementation* dalyje — metodo realizavimui reikalingą ruošinį. Metodas užrašomas kaip *Object Pascal* procedūra.

Pavyzdžiui, pasirinkus Form1 įvykį *OnClick*, kuriamas metodas *FormClick*, kuriuo programos vykdymo metu reaguojama į spragtelėjimą pele pagrindinės formos lange. Metodo antraštė:

```
procedure FormClick(Sender: TObject);
implementation dalyje atsiranda tokios eilutės:
procedure TForm1.FormClick(Sender: TObject);
begin
end;
```

Programos kūrėjui tarp *begin...* ir *end* reikia užrašyti reakcijos veiksmus. Tarkime, paspaudus pelę, formos langas turi būti nudažomas raudonai. Tam šioje vietoje reikia įrašyti eilutę:

```
Form1.Color := clRed;
```

Tai priskyrimo sakiny, kuriuo Form1 objekto savybei *Color* suteikiama nauja reikšmė *clRed*.

Panašiu priskyrimo sakiniu galima keisti ir kitų Form1 savybių, ir kitų objektų savybių reikšmes:

```
ObjektoVardas.SavybėsVardas := NaujaReikšmė;
```

Su programos objektais galima atlikti tam tikrus veiksmus (vykdyti objektų metodus). Objektai, neturintys jokių metodų, yra beprasniai. Standartinių *Delphi* komponentų metodai realizuoti pačioje sistemoje. Bendru atveju daugumos metodų vykdymo sakinių struktūra yra tokia:

```
ObjektoVardas.MetodoVardas (Parametrai);
```

arba (jei metodas neturi parametru):

```
ObjektoVardas.MetodoVardas;
```

Pavyzdžiui, sakiny

```
Form1.Close;
```

užveria Form1 langą ir baigia programos darbą.

Bet kurio programoje panaudoto standartinio komponento savybių ir metodų sąrašus galima pamatyti, objekto viduje paspaudus *Ctrl+F1*. Tą pačią informaciją galima susirasti ir pasinaudojus *Delphi* pagalbos priemonėmis (*Help* komanda pagrindiniame *Delphi* lange).

Į TForm1 klasę galima įtraukti ne tik reakcijos į įvykius metodus, bet ir sukurti savus. Tokie metodai taip pat užrašomi kaip *Object Pascal* procedūros ar funkcijos. Šiuos metodus aprašo programuotojas.

Programų komentarai

Nors programos veiksmus atlieka ne žmogus, bet kompiuteris, tačiau dėl daugelio priežasčių programų kūrėjams dažnai tenka skaityti ir keisti programų tekstus. Todėl programos rašomos taip, kad būtų lengva jas skaityti: paliekami papildomi tarpai, tuščios eilutės, tekstas atitraukiamas nuo kairiojo krašto ir pan. Kartu programose labai naudingi paaiškinimai, padedantys geriau

suprasti panaudotas duomenų struktūras ir atliekamus veiksmus. Tokie paaiškinimai (komentarai) programose *Delphi* terpėje *Object Pascal* kalba gali būti užrašyti dviem būdais.

Pirmasis būdas paprastai taikomas trumpiems komentarams. Šiuo atveju jie pradedami dviem pasvirusiais brūkšneliais (//). Kompiuteris, kurioje nors programos eilutėje radęs //, už šių ženklų iki eilutės galo esantį tekstą ignoruoja ir kompiliavimo metu nenagrinėja.

Komentarai, kuriuos reikia užrašyti keliose greta esančiose eilutėse, pradedami ir užbaigiami riestiniais skliaustais. Be riestinių skliaustų, galima naudoti ir paprastus skliaustus kartu su žvaigždutėmis: (* ir *). Aišku, taip galima užrašyti ir vienos eilutės komentarą.

Komentarai tarp riestinių skliaustų kompiliavimo metu taip pat yra ignoruojami, išskyrus tuos atvejus, kai po riestinio skliausto iš karto yra dolerio ženklas. Riestiniu ir dolerio ženklais pradamos direktyvos (nurodymai) kompiliatoriui. Nesudėtingoms mūsų programoms jų nereikia, todėl plačiau apie jas nerašysime.

Eilutės su komentarais pavyzdys:

```
Form1.Color:= clRed; // nudažoma raudonai  
arba  
Label1.Font.Size:= 20; { nustatomas naujas šrifto dydis — 20 pt }
```