

Delphi

Delphi je programski paket namenjen kreiranju Windows aplikacija (programa) za personalne računare. Krajnji proizvod Delphi-ja je izvršni program koji rešava zadatke zbog kojih je kreiran u okruženju operativnog sistema Windows. Programski jezik koji koristi Delphi je modernizovana verzija Pascal-a poznata pod nazivom *Object Pascal*.

Delphi spada u **objektno-orijentisane programe** (OOP). Objektno orijentisano programiranje omogućuje kreiranje komponenti za višestruko korišćenje u raznim korisničkim programima. Osnovni pojmovi OOP su: komponente, objekti, metode, svojstva i događaji.

Komponente su elementi ugrađeni u Delphi-ju za višestruku upotrebu i koriste se pri formiranju objekta.

Objekt je osnovni element Delphi programa koji ima odgovarajuća svojstva (karakteristike), metode koje definišu njegovo ponašanje i mogućnost prepoznavanja događaja na koje mogu reagovati.

Metod je programski kod pridružen objektu, koji određuje kako de objekt manipulirati sa informacijama i kako de reagovati na događaje.

Svojstva su karakteristike objekata kao što su: veličina, položaj, boja ili font teksta. Svojstva određuju pojavljivanje, a ponekad i ponašanje objekta.

Događaji mogu biti prouzrokovani od strane korisnika ili od strane operativnog sistema Windows.

NAPOMENA: U ovom priručniku korišćene su verzije: 6, 7, 2005 i 2006 DELPHI-ja.

Početak rada i upravljanje Delphi-jem

Kada se pokrene program ekran dobija izgled kao na slici (slika 1). Polazni ekran ima 8 delova:

1. naslovnu liniju,
2. meni liniju,
3. liniju alata,
4. liniju komponenti,
5. prozor za hijerarhijski prikaz komponenti aplikacije,
6. prozor za određivanje osobina i događaja objekta u aplikaciji,
7. prozor obrasca za dizajniranje aplikacije,
8. prozor za pisanje programskog koda.

Naslovna linija sadrži ikonicu kontrolnog menija programa Delphi, naziv projekta koji se realizuje i tri dugmeta za upravljanje veličinom prozora aplikacije.

Meni linija je klasična Windows meni linija na kojoj se nalazi 11 opcija koje se aktiviraju klikom miša na željenu opciju.

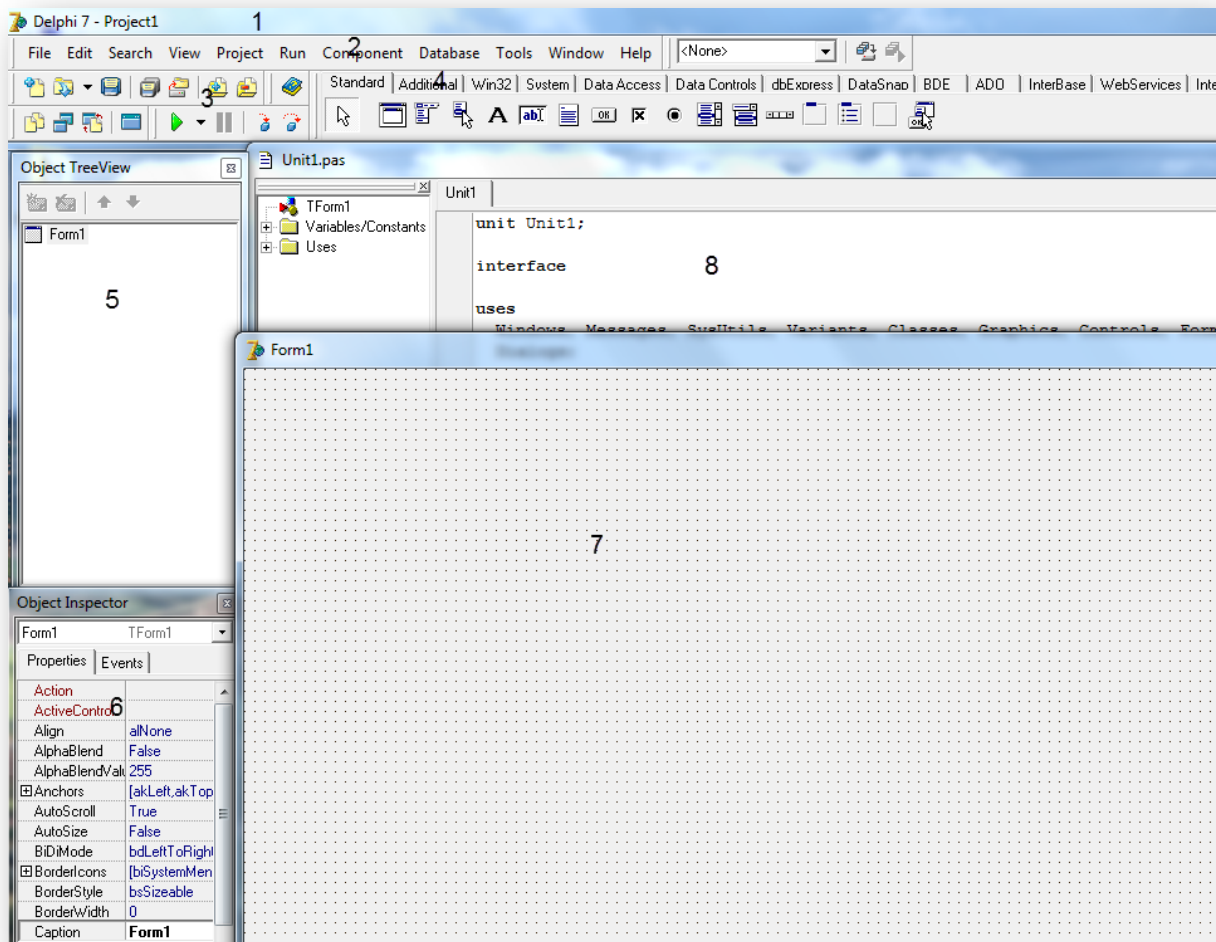
Linija alata sadrži najčešće korišćene komande koje se zadaju direktnim klikom miša na odgovarajuću sličicu.

Linija komponenti sastoji se od više stranica, pri čemu svaka stranica ima svoje komponente.

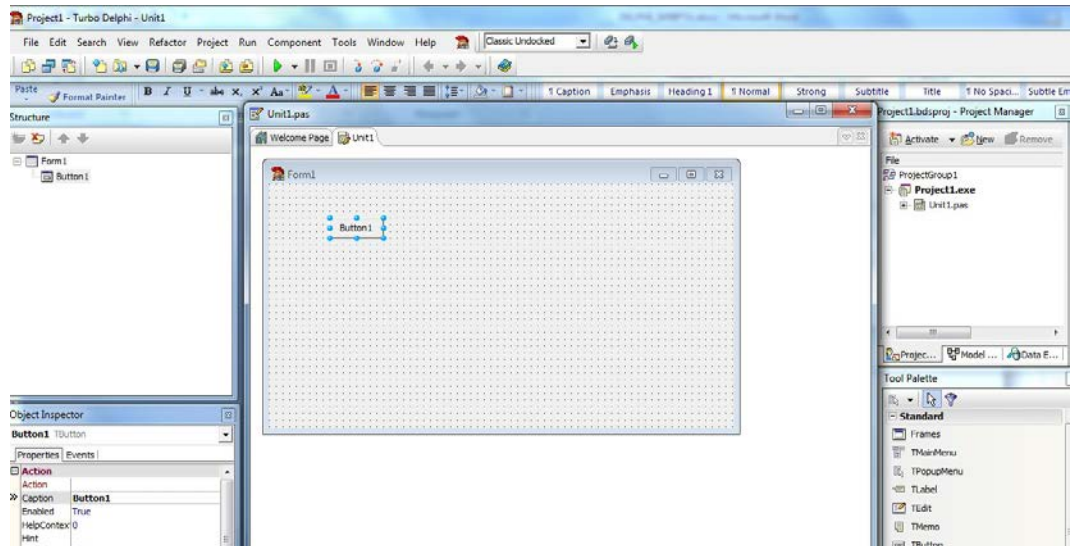
U **prozoru za hijerarhijski prikaz komponenti aplikacije** (*Object TreeView*) se prikazuje hijerarhijsko stablo objekata postavljenih u okviru aplikacije.

Prozor za određivanje osobina objekta i događaja (*Object Inspector*) se nalazi na levoj strani ekrana i služi za podešavanje osobina (*Properties*) i događaja (*Events*) objekata koje se formiraju tokom razvoja aplikacije.

Prozor obrasca za dizajniranje automatski se formira pri pokretanju Delphi-ja i naziva se forma. Forma generički dobija ime **Form1**. Ova forma iako prazna poseduje osobine Windows programa: može se premeštati, mogu joj se menjati dimenzije, može se minimizirati, maksimizirati i zatvoriti. Ima i svoju ikonicu koja se dodeljuje pri minimizaciji, ima naslovnu liniju, okvir i radnu površinu. Na radnoj površini nalazi se mreža tačkica koja služi kao pomod pri postavljanju i poravnavanju komponenti.



Slika 1




Slika 2

Obrasci (Forme)

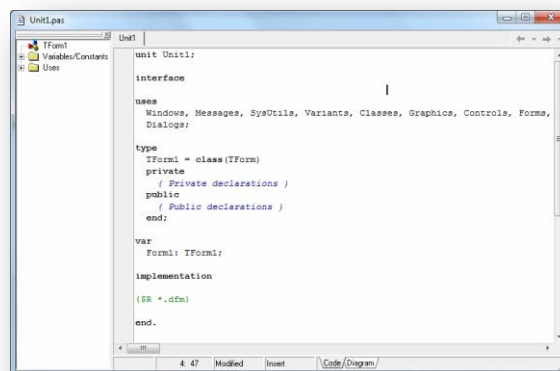
Obrasci (slika 1) su skice prozora za vreme projektovanja. Služe kao nosači komponenta (vlasnici su kompenenata). Komponente ne mogu da postoje izvan obrazaca.

Jedan projekat može sadržati više obrazaca. Nov obrazac se dodaje projektu komandom **File/New/Form** ili uz pomod dugmeta prečice **New Form** na paleti komandi.

Manipulacija sa obrascima najlakše se vrši uz pomod **Project Managera** (komandom Delphi-ja *View/Project Manager*).

Startovanje programa se realizuje preko menija izborom **Run** pa zatim iz podmenija opet **Run** ili pomodu ikone sa zelenom strelicom iz palete ispod menija . Povratak nazad (izlazak iz aplikacije) se realizuje preko dugmeta za zatvaranje prozora (X). Forma predstavlja platformu za dodavanje komponenti i razvoj aplikacije.

Praznoj formi se automatski dodeljuje i modul koji dobija ime Unit1 i koji sadrži definiciju forme u tipu TForm1. Unit1 ima poseban prozor za editovanje (urednik koda) u kome se pišu naredbe Pascala koje rešavaju dati zadatak. Ovom modulu se može pristupiti preko jezička Code koji se nalazi uz donju ivicu radnog prozora. Povratak na formu se realizuje preko jezička Design.



Slika 3

Svojstva obrasca

Najvažnija svojstva obrasca su: *Caption* (natpis), *Name* (ime), *Font* (Boja veličina, stil), *Color* (boja), *Hint* (objašnjenje), *Height* (visina), *Width* (širina), *Cursor* (izgled kursora).

Obrasci imaju relativno veliki broj specifičnih svojstava. Važnija svojstva su:








1. **FormStyle** (stil obrasca)-označava vrstu obrazaca:
 - ***fsNormal*** (normalan) označava uobičajenu vrstu obrazaca.
 - ***fsStayOnTop*** (ostani na vrhu) označava prozore koji se ne mogu pokriti drugim prozorima iste aplikacije.
 - ***fsMDIForm*** (MDI obrazac) označava prozor roditelja koji sadrži druge obrasce.
 - ***fsMDIChild*** (MDI dete) označava prozor dete koji se smešta u radni prostor prozora roditelja.
2. **BorderIcons** (ivične sličice)-određuje koja de se dugmad nadi pored trake sa natpisom prozora. Elementi skupa mogu da budu *biSystemMenu*, *biMinimize*, *biMaximize* i *biHelp*.
3. **BorderStyle** (stil ivica)-određuje način iscrtavanja ivica prozora:
 - ***bsSizeable*** (podesive)- veličinu prozora moguće je menjati povlačeći ivice pomodu miša.
 - ***bsSingle*** (jednostruke-nepodesive)- nemoguće je menjati pomodu miša.
 - ***bsSizeToolWin*** (podesive za prozore sa alatkama)- veličinu prozora moguće menjati pomodu miša. Traka sa natpisom manja i slova sitnija. Može da ima samo dugme za zatvaranje prozora. Ovakvi prozori namenjeni su pravljenju paleta sa alatkama.
 - ***bsToolWindow*** (nepodesive za prozore sa alatkama)- kao u slučaju *bsSizeTool* samo nemoguće menjati veličinu prozora pomodu miša.
 - ***bsDialog*** (za dijalog)- namenjen prozorima za dijalog. Nema dugme za minimiziranje i maksimiranje a obavezno ima dugme za zatvaranje i može dugme za pomod.
 - ***bsNone*** (bez)- prozor bez ivica, trake sa natpisom i upravljačke dugmadi. Ne može da se menja veličina prozora ni da se pomera.
4. **Position** (položaj)-određuje mesto i veličinu prikazivanja prozora prilikom izvršavanja aplikacije.
5. **WindowState** (stanje prozora) označava trenutno stanje prozora na ekranu. Vrednost *wsMaximized* označava da je prozor pretvoren u najveću moguću veličinu, *wsMinimized* da je pretvoren u minimalan oblik, a *wsNormal* da prozor nije ni u jednom od prethodna dva stanja.
6. **AutoScroll** (automatsko prelistavanje)- *HorzScrollBar* (vodoravni klizač) i *VertScrollBar* (uspravni klizač), određuju ponašanje vodoravnog i uspravnog klizača.
7. **Icon** (sličica)- označava sličicu na upravljačkom dugmetu prozora.
8. **KeyPreview** (prethodni pregled tastera)- određuje način obrade događaja u vezi s tastaturom (*OnKeyDown*, *OnKeyUp*, *OnKeyPress*). Vrednost **True** označava da se prvo poziva odgovarajući rukovalac događajima obrasca a tek posle rukovalac događaja

komponente. Vrednost **False** označava da se pozivaju samo rukovaoci događajima komponentata.

9. **ActiveControl** (aktivna upravljačka komponenta)- označava koja je komponenta u obrascu trenutno u žiži.
10. **WindowMenu** (meni za prozore)- označava stavku menija čiji je podmeni u MDI aplikaciji predviđen za upravljanje prozorima decom.
11. **Active** (aktivan)- s vrednošću *True* označava da je prozor trenutno aktivan, dok vrednost *False* označava da nije. Ovo svojstvo je raspoloživo samo za vreme izvršenja projekta i ne može da se menja iz programa.
12. **ModalResult** (uslovljeni rezultat)- određuje način zatvaranja uslovljenog prozora (najčešće prozora za dijalog). Podrazumevana vrednost je 0. Kada se promeni vrednost u nenultu, prozor izgubi uslovljeni status i zatvara se. Simboličke konstante za tipične namene su: *mrNone* (ništa), *mrOK* (sve je u redu), *mrCancel* (odustani), *mrAbort* (prekini), *mrRetry* (pokušaj ponovo), *mrIgnore* (zanemari), *mrYes* (da), *mrNo* (ne), *mrAll* (da za sve).
13. **AlphaBlend** (alfa mešanje)- S vrednošću *True* označava da se prozor iscrtava prozirno, s vrednošću *False* da se prozor ne iscrtava prozirno.
14. **AlphaBlendValue** (vrednost alfa mešanja)- određuje stepen prozirnosti iscrtavanja prozora (0-jasna slika, 255-slika se uopšte ne nazire).
15. **MDIChildCount** (broj MDI dece)- predstavlja broj obrazaca dece u radnom prostoru posmatranog obrasca. Ovo svojstvo je raspoloživo samo za vreme izvršavanja aplikacije.
16. **MDIChildren** (MDI deca)- je niz obrazaca u posmatranom obrascu. Prvi u nizu ima indeks 0. Ovo svojstvo je raspoloživo samo za vreme izvršavanja aplikacije.
17. **ActiveMDIChild** (aktivno MDI dete)- predstavlja obrazac koji je trenutno u žiži unutar posmatranog obrasca. Ovo svojstvo je raspoloživo samo za vreme izvršavanja aplikacije.
18. **TileMode** (način raspoređivanja pločica)- određuje na koji de se način rasporediti prozori deca u radnom prostoru prozora roditelja.

Metode obrasca








Osim metoda koje ima i vedina komponenta, važnije metode su:

-  **constructor *Create*** -stvari. Ova metoda je konstruktor za sve komponente pa i za sve obrasce.
-  **procedure *Release*** -oslobodi. Ovom metodom se uništava obrazac i oslobađa memorija koju on zauzima.
-  **procedure *Show*** -prikaži. Ova metoda na neuslovljen način čini da obrazac postane vidljiv (u žiži). To postiže postavljanjem vrednosti svojstava *Visible* na *True*.
-  **function *ShowModal*** –prikaži uslovljeno. Ova metoda na uslovljen način čini da obrazac postane vidljiv. Iz ove funkcije se vrada tek pošto se prozor zatvori.
-  **procedure *Hide*** –sakrij. Ovom metodom obrazac postaje nevidljiv. To postiže postavljanjem vrednost svojstava *Visible* na *False*.
-  **procedure *ScrollInView*** -pomeri u vidno polje. Ova metoda potpuno ili delimično dovodi komponentu ili prozor dete u vidno polje prozora.
-  **procedure *Close*** –zatvori.

-  procedure **Next** –slededi.
-  procedure **Previous** –prethodni.

Događaji obrasca

Osim događaja koji mogu da se dešavaju i komponentama važniji događaji su:

-  Događaj **OnCreate** (prilikom stvaranja) nastaje u toku stvaranja obrasca metodom Create.
-  Događaj **OnActivate** (prilikom aktiviranja) nastaje kada se žiža pomeri na prozor.
-  Događaj **OnShow** (prilikom prikazivanja) nastaje pre nego što se prozor prikazuje.
-  Događaj **OnResize** (prilikom promene veličine) nastaje kada se menja veličina obrasca.
-  Događaj **OnHide** (prilikom sakrivanja) nastaje pre uklanjanja prozora sa ekrana.
-  Događaj **OnDestroy** (prilikom uništavanja) nastaje pre uništavanja obrasca.
-  Događaj **OnCloseQuery** (prilikom upita zatvaranja) nastaje kada je započeta akcija za zatvaranje.

Prazan projekat mozemo dobiti na jedan od slededih načina: pokretanjem programa Delphi ili korišćenjem opcije **File/New/Application** s meni linije.

Ako pogledamo linju poslova nakon pokretanja praznog projekta, videdemo da kao ime projekta stoji **Project1**. Da bismo sačuvali aplikaciju moramo uraditi sledede:

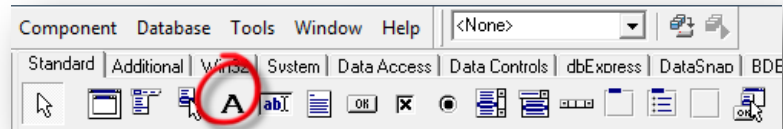
1. Na meni liniji odabrati opciju **File/Save All....** nakon čega se dobija dijalog prozor.
2. U polju za ime datoteke treba uneti dozvoljeno ime i kliknuti na dugme **Save**. Ova datoteka je Pascal datoteka.
3. U slededem koraku pojavljuje se komunikacioni prozor u kome treba zadati ime projekta.

Ukoliko je upravo snimljen projekat zatvoren, da bi se ponovo otvorio treba sprovesti slededu proceduru:

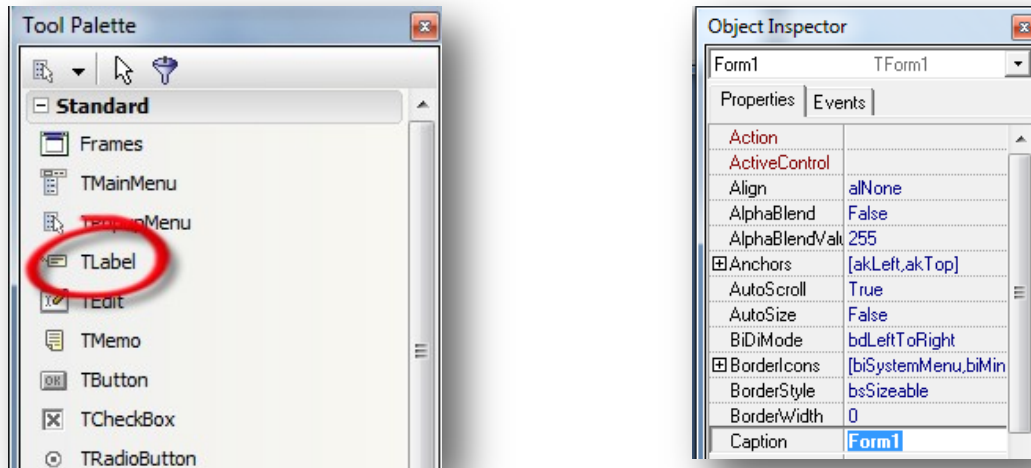
1. Na meni liniji odabrati opciju **File/Open Project....Ctrl+F11**.
2. Željeni projekat treba selektovati i nakon toga treba aktivirati dugme **Open**.

Prvi program

Paleta gotovih komponenti koje se mogu dodavati formi (aplikaciji) se nalazi uz gornju ivicu radnog prozora (verzija 7) ili sa leve strane (verzija 2005 i 2006). Na početku liste je grupa **Standard**, zatim sledi grupa **Additional** itd. Imena komponenti počinju slovom T (verzija 2005 i 2006 slika 4) (TLabel, TEdit, TButton, ...), a kod verzije 7 nema prefiksa T(slika 3). Dodavanje komponenti programu se realizuje tako što se klikne mišem na komponentu u paleti a zatim na formu. Komponenta se automatski dodaje formi sa preddefinisanim imenom ali bez slova T (Label1, Label2, ..., Edit1, Edit2,...) Komponentama se podešavaju svojstva u Inspektoru objekata. Inspektor objekata ima dve kartice - **Properties (svojstva)** i **Events (događaji)**.

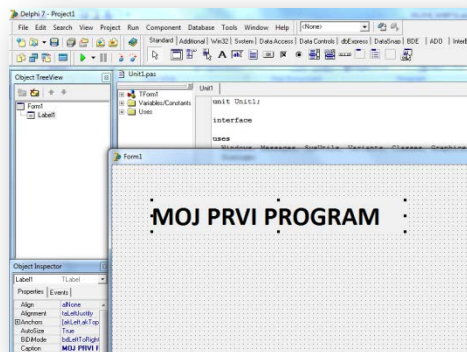


Slika 4



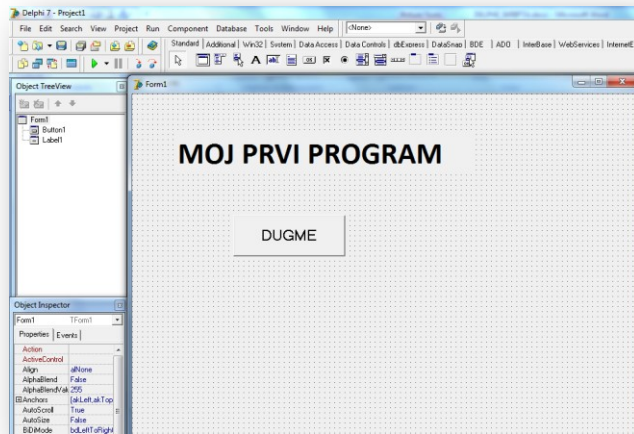
Slika 5

Prvi program de ispisati neku poruku (npr. 'Moj prvi program'). U tu svrhu treba dodati komponentu **Label**. Odaberemo iz palete komponentu obeleženu sa **TLabel** i zatim kliknemo na formu. Na formi de se pojaviti tekst **Label1** sa plavim markerima oko teksta. Promenidemo tekst **Label1** u tekst **Moj prvi program**. Promena teksta znači promenu jednog svojstva komponente **Label**. Potrebno je u Inspektoru objekata pronadi svojstvo **Caption**. Desno od naziva svojstva u polju za editovanje piše **Label1**. Treba kliknuti mišem u to polje i upisati novi tekst **Moj prvi program**.



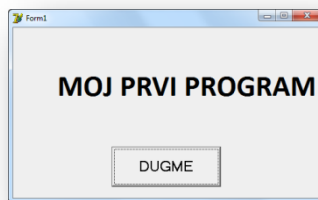
Slika 6

Dodademo formi i komponentu **Button** (dugme). Kliknite na komponentu **Button** i zatim na formu na mesto gde bi trebalo da bude postavljeno dugme (Slika 7). Komponente se mogu pomerati i preuređivati na formi - pomod predstavlja mreža vođica.



Slika 7

Startujte program (**Run** pa zatim opet **Run** ili zelena strelica iz palete ispod menija). Dobide se prozor našeg programa.



Slika 8

Pri zatvaranju prozora Delphi-ja, ako nije snimljen projekat (aplikacija) dobide se upozorenje sa pitanjem da li treba sačuvati projekat.

Vežba

Kreirati aplikaciju koja de imati jedan obrazac (formu) crvene bije u trenutku aktivacije, širinu 600, visinu 400 i koja de na jedan klik na formu da promeni boju forme u belo a na dvoklik da poveda visinu na 500 i širinu 750.

```
procedure TForm1.FormActivate(Sender: TObject);
```

```
begin
```

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

```
end;
```

```
procedure TForm1.FormClick(Sender: TObject);
```

```
begin
```

```
Form1.Color:=clWhite;
```

```
end;
```



```

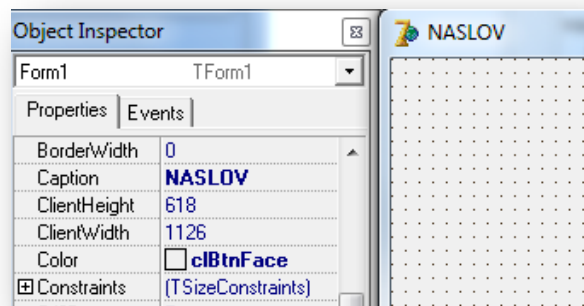
procedure TForm1.FormDbClick(Sender: TObject);
begin
  Form1.Height:=500;
  Form1.Width:=750;
end;

```

Svojstva komponenti

U Inspektoru objekata vrši se podešavanje svojstava (osobina) komponenti. Neka svojstva podešavamo tako što biramo između ponuđenih opcija a neka svojstva upisujemo preko tastature.

Podesidemo nekoliko svojstava forme. Svojstvo **Caption** je inicijalno podešeno na Form1. Kliknimo na naziv svojstva u edit polju desno od naziva i upišimo **Ovo je Caption (Title bar)**. U naslovu forme se pojavljuje upisani tekst.



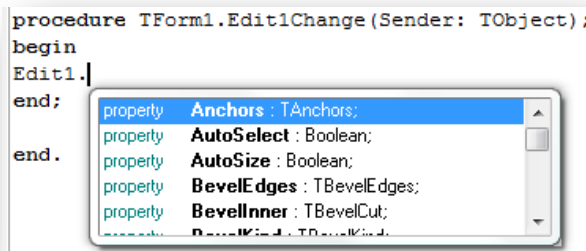
Slika 9

Svojstvo **Font** se koristi za podešavanje fonta koji de biti primenjen na sve komponente koje se budu dodavale (za svaku komponentu se može podesiti poseban font). Kada se klikne na svojstvo **Font** desno se pojavljuje dugme sa tri tačke. Ako kliknemo na to dugme pojavi se dijalog prozor u kome podešavamo svojstvo (za font se dobija klasičan Windows dijalog). Ako ispred naziva svojstva stoji znak plus (+) onda to znači da klikom na to plus otvaramo podsvojstva koja zatim možemo podesiti. Kod svojstva font se pojavljuju podsvojstva **Charset, Color, Size, ...**. Podesite veličinu (Size) fonta na 10.

Svojstvo **BorderIcons** se koristi za podešavanje kontrola prozora aplikacije (forme). Podsvojstva imaju vrednosti **True** (svojstvo je uključeno) i **False** (svojstvo je isključeno). Na osnovu naziva svojstva može se zaključiti koja mu je funkcija - najbolje je proveriti izborom i startovanjem programa. *Ako ste podsvojstvo **BiSystemMenu** podesili na False i startovali program nedete modi prekinutu program jer su isključena kontrolna dugmad. U tom slučaju treba odabrati iz glavnog menija Run pa iz podmenija **Program reset** ili iskoristiti Windows-ovu kombinaciju tastera Alt-F4.* Na isti način se podešavaju i svojstva komponenti koje se dodaju na formu.

Kako pamtiti (i da li treba) sve osobine komponenti? U Inspektoru objekata se može uvek proveriti koje osobine poseduje koja komponenta, ali ako se piše kôd nezgodno je stalno se prebacivati na prozor sa osobinama i natrag na prozor sa kodom. Delphi nudi pomod u obliku

tzv. **IntelliSense** tehnologije. IntelliSense ispisuje listu osobina komponente koja se koristi. Strelicama na gore i na dole bira se osobina a pritiskom na taster Tab dodaje kôdu.



Slika 10

Zahvaljujuci IntelliSense-u ne moraju se pamtititi sve osobine svih komponenti.

Metode komponenata

A. constructor **Create** –stvari

Ova metoda je konstruktor za sve komponente.

Po stvaranju komponente treba podesiti svojstva te komponente ili naredbom ili odgovarajućom metodom (posebno svojstvo **Name**). Da bi nova komponenta bila upotrebljiva treba dodeliti vrednost svojstvu **Parent** (roditelj). Najčešće se navede vrednost **Self** čime se tekuci obrazac proglašava vlasnikom. Primer:

```
NovoDugme:=TButton.Create(Self);
```

```
NovoDugme.Parent:=self;
```

B. procedure **Free** –oslobodi

Ovom metodom uništava se komponenta i oslobađa memorija.

C. procedure **Show** -prikaži

Ova metoda čini da tekuda komponenta postane vidljiva i to postiže postavljanjem vrednosti svojstva **Visible** na True.

D. procedure **Hide** -sakrij

Ova metoda čini da komponenta postane nevidljiva postavljanjem vrednosti **Visible** na False.

E. procedure **SetBounds** (Levo, Gore, Širina, Visina: Integer); -postavi granice

Ova metoda određuje položaj i veličinu komponente odjednom.

Događaji komponenata

I. Događaji pri pomeranju žiže

Kada se žiža pomeri na neku komponentu, deside joj se događaj **OnEnter** (prilikom ulaska), a kada se žiža pomeri na drugu komponentu, događaj **OnExit** (prilikom izlaska).

II. Događaji sa mišem

Najčešće korišćen događaj pomodu miša jeste pritisak na komponentu mišem **OnClick** (prilikom pritiska mišem).

Drugi često korišćen događaj je dvostruki pritisak komponente mišem **OnDbClick** (prilikom dvostrukog pritiska mišem).

Slededa dva događaja pomodu miša jesu pritiskanje nekog tastera miša (bez otpuštanja) **OnMouseDown** (prilikom pritiskanja miša) i **OnMouseUp** (prilikom otpuštanja miša).

III. Događaj sa tastaturom

Osnovni događaj koji može da se proizvede pomodu tastature jeste pritisak i otpuštanje nekog tastera na tastaturi **OnKeyPress** (prilikom pritiska na taster) i **OnKeyUp** (prilikom otpuštanja tastera).

KONVERZIJE

Konverzije se koriste za pretvaranje jednog tipa podatka u neki drugi tip.

StrToInt() – pretvara string u integer

StrToFloat() – pretvara string u realan broj

IntToStr() – pretvara integer u string

FloatToStr() – pretvara realan broj u string

TimeToStr(Time) – pretvara vreme u string

DateToStr(Date) – pretvara datum u string

Za pretvaranje znakovnog niza u numerički tip i obrnuto se može koristiti i funkcija **val**. Oblik funkcije **val** je slededi:

val(s, x, i)

gde je s tekst (string) koji je unešen u neko edit polje i treba da se pretvori u broj, x je brojeva promenljiva koja de prihvatiti tu vrednost, a i je indikator uspešnosti pretvaranja. Ako ima vrednost 0, pretvaranje je uspešno izvršeno.

Za pretvaranje brojevine vrednosti u tekst koristi se slededa funkcija:

str(x,s),

gde su x i s isto što i u funkciji val. Kod ove funkcije je nepotreban indikator uspešnosti.

Jednostavne komponente

Delphi nudi ogroman broj komponenata za sastavljanje prozora aplikacije. Jednostavne komponente se koriste samostalno, bez saradnje sa drugim komponentama. To su istovremeno i komponente koje se najčešće i u najvedem broju sredi u svakodnevnoj praksi.

Prikazivanje teksta

Za prikazivanje teksta u prozoru koriste se specijalne komponente čiji sadržaj korisnik ne može neposredno da menja mišem ili preko tastature za vreme izvršavanja aplikacije. Koriste se za obeležavanje namene komponenata i za prikazivanje raznih informacija (rezultata) koje se dobijaju u toku izvršavanja aplikacije.

A

Label (Oznaka)

To je komponenta bez prozora i ne može biti u žiži. Može da joj se dešava vedina događaja koja se proizvode mišem dok događaji proizvedeni tastaturom ne mogu da joj se dese, jer za to komponenta mora da bide u žiži.

Oznaka je komponenta tipa *TLabel* i za njeno umetanje u obrazac za vreme projektovanja postoji dugme prečica **Label**, na stranici **Standard**, na paleti komponenata u glavnom prozoru Delphija.

- Svojstvo **Caption** (natpis) –tipa *String* glavno je svojsvo oznake;
- Svojstvo **Name** (ime) –tipa *String*;
- Svojstvo **Font** (font) –podešava se u dijalog prozoru;
- Svojstvo **Color** (boja) –tipa nabiranja;
- Svojstvo **AutoSize** (automatska veličina) –tipa *Boolean* s podrazumevanom vrednošću *True* označava da se veličina oznake automatski menja kad god se promeni sadržaj svojstva **Caption**, vrednost *False* označava da se veličina komponente ne menja;
- Svojstvo **Alignment** (poravnanje) –tipa nabiranja određuje način poravnanja natpisa po širini oznake;
- Svojstvo **Layout** (raspored) –tipa nabiranja određuje poravnanje natpisa po visini oznake;
- Svojstvo **WordWrap** (prebacivanje reči) –tipa *Boolean* s vrednošću *True* označava da se tekst natpisa prelama u više redova, vrednost *False* označava da se tekst prikazuje u jednom redu;
- Svojstvo **ShowAccelChar** (prikaži znak za prečicu) –tipa *Boolean* s vrednošću *True* označava da znak & obeležava znak prečicu i ne treba ga prikazivati na ekranu, vrednost *False* označava da znak & treba prikazivati i da oznaka nema znak prečicu.

**Button (dugme)**

Osnovna uloga dugmeta jeste da se pritiskom na njega pokrene neka akcija.

Za umetanje dugmeta u obrazac postoji dugme prečica **Button** na stranici **Standard**, na paleti komponenata.

- Svojstvo **Caption** (natpis) –tipa *String* služi za obeležavanje namene dugmeta;
- Svojstvo **Name** (ime) –tipa *String*;
- Svojstvo **Font** (font) –podešava se u dijalog prozoru;
- Svojstvo **Cancel** (odustani) –tipa *Boolean* s vrednošću *True* označava da je dato dugme za odustajanje, vrednost *False* označava da nije dugme za odustajanje;
- Svojstvo **ModalResult** uslovljeni rezultat –tipa *Integer* određuje način zatvaranja uslovljenih prozora.

**BitBtn (dugme sa slikom)**

Dugme sa slikom ima istu namenu i koristi se na isti način kao i obično dugme. Jedina razlika je u tome što pored natpisa može da se obeleži i nekom slikom.

Za njegovo umetanje u obrazac postoji dugme prečica **BitBtn**, na stranici **Additional**, na paleti komponenata.

Svojstva kao kod običnog dugmeta i još:

- Svojstvo **Kind** (vrsta) tipa nabiranja određuje vrstu dugmeta.
- Svojstvo **Glyph** (gravura) tipa *TBitmap* predstavlja sliku na dugmetu.
- Svojstvo **NumGlyph** (broj gravura) tipa *Integer* onačava na koliko delova treba podeliti po širini sadržaj svojstva **Glyph**.
- Svojstvo **Layout** (raspored) tipa nabiranja određuje relativan položaj slike i teksta na dugmetu.



Bevel (ram)

Ram je komponenta bez prozora koja služi za vizuelno grupisanje komponenata. Nije kontejnerska komponenta i nikakvi događaji ne mogu da mu se dešavaju.

Za njegovo umetanje u obrazac postoji dugme prečica **Bevel**, na stranici **Additional**, na paleti komponenata.

- Svojstvo **Shape** (oblik) tipa nabiranja određuje vrstu rama.
- Svojstvo **Style** (stil) tipa nabiranja određuje da li se prikazuje upušteno (*bsLowered*) ili izdignuto (*bsRaised*).



Timer (časovnik)

Časovnik je nevizuelna komponenta pomodu koje može da se meri vreme.

Časovnik periodično generiše događaj **OnTimer** (prilikom isteka vremena) kad god istekne zadati period. To omogućava da program u tim trenucima nešto uradi, a da korisnik ne preduzme nikakvu akciju.

Za njegovo umetanje u obrazac postoji dugme prečica **Timer**, na stranici **System**, na paleti komponenata.

- Svojstvo **Interval** tipa *Integer* određuje koliko često de se generisati događaji *OnTimer*.
- Svojstvo **Enabled** tipa *Boolean* određuje da li časovnik radi, podrazumevana vrednost *True* označava da časovnik radi, a *False* da je časovnik zaustavljen.



Edit (okvir za tekst)

Okvir za tekst omogućuje unošenje jednog reda teksta preko tastature i numeričkih odataka.

Za njegovo umetanje u obrazac za vreme projektovanja postoji dugme prečica **Edit**, na stranici **Standard**, na paleti komponenata.

- Svojstvo **Text** (tekst) –tipa *String* je glavno svojstvo koje sadrži tekst u posmatranom okviru. Njegova vrednost može da se postavi za vreme projektovanja i da se promeni za vreme izvršavanja aplikacije.
- Svojstvo **CharCase** (veličina slova) –tipa nabiranja određuje da li se u tekstu prihvataju mala i velika slova onako kako se unose preko tastature.
- Svojstvo **PasswordChar** (znak za lozinku) –tipa *Char* određuje znak koji se na ekranu prikazuje u okviru za tekst kada se unosi lozinka.



Memo (višeredni okvir za tekst)

Višeredni okvir za tekst omogućuje unošenje proizvoljnog broja redova teksta preko tastature. Za njegovo umetanje u obrazac postoji dugme prečica **Memo**, na stranici **Standard**, na paleti komponenata.

- Svojstvo **Lines** (redovi) tipa *TStrings* je glavno svojstvo koje sadrži tekst uposmatranom višerednom okviru za tekst.
- Svojstvo **Aligment** (poravnanje) tipa nabiranja određuje način poravnanja teksta po širini.
- Svojstvo **ScrollBars** tipa nabiranja određuje koli klizači postoje.
- Svojstvo **Word Wrap** (prebacivanje reči) tipa *Boolean* s podrazumenaom vrednošću *True* označava da se redovi automatski prelamaju kada se tekst poravnava uz levu ivicu, vrednost *False* označava da se redovi ne prelamaju.

Dodavanje komponenti

Dodavanje komponenti programu se realizuje tako što se klikne mišem na ime komponente u paleti komponenti a zatim na formu. Komponenta se automatski dodaje formi sa preddefinisanim imenom ali bez slova T (Label1, Label2, ..., Edit1, Edit2,...) i preddefinisanim svojstvima.

Komponenta Label

Pomodu ove komponente mogu se ispisivati poruke i rezultati obrade podataka.

Primer. Kliknite dva puta na labelu koju ste dodali formi. Pojavide se urednik koda. Između **begin** i **end;** Ukucajte naredbu **Label1.Caption:='Dobar dan. Hvala sto ste kliknuli na mene';**.

```
Unit1
procedure TForm1.Label1Click(Sender: TObject);
begin
    Label1.Caption:='Dobar dan. Hvala sto ste kliknuli na mene';
end;
end.
```

Slika 11

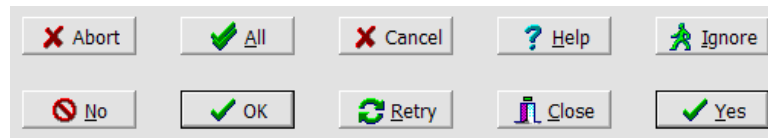
Između apostrofa možete ukucati i tekst po želji. Startujte program i kliknite na tekst **Label1**. Programski smo uticali na svojstvo `Caption` labele. Dodajte još dve naredbe:

```
Label1.Font.Size:=20;
Label1.Font.Color:=clRed;
```

Startujte program i kliknite na labelu.

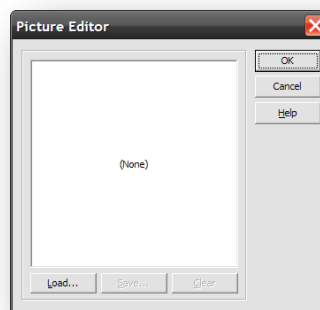
Komponente **Button** i **BitBtn**

Komponente **Button** i **BitBtn** (dugmad) omogućavaju da se klikom na njih pokrene neka akcija. Razlika između ove dve komponente je u tome što pored natpisa komponenta **BitBtn** može da se obeleži i nekom slikom radi lakšeg prepoznavanja. Svojstvo **Kind** o dređije vrst u `du gmeta`. Na slici su data sva preddefinisana **BitBtn** dugmad.



Slika 12

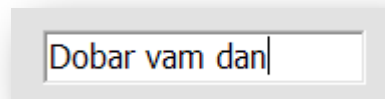
Vrednost `bkCustom` svojstva `Kind` označava dugme kome je sliku pridružio programer. Slika se dodaje preko svojstva **Glyph**. Klikom na dugme sa tri tačke aktivirade se urednik slika (Picture Editor).



Slika 13

Komponenta **Edit**

Komponenta **Edit** (okvir za tekst ili tekst boks) omogućava unos jedne linije teksta preko tastature. Može se uneti najviše 64K znakova.



Slika 14

Uneti tekst je sadržan u svojstvu **Text** tipa `String`. Pomodu ove komponente se mogu unositi i numerički podaci (brojevi), ali su potrebne konverzije. Za konverziju teksta u ceo broj koristi se

funkcija **StrToInt**, a za pretvaranje teksta u realan broj koristi se funkcija **StrToFloat**. Za konverzije broja u tekst (string) koriste se funkcije **IntToStr** i **FloatToStr**.

Primer Izračunati zbir dva broja.

Neka je za prvi broj određeno Edit1 a za drugi broj Edit2 polje i neka je za ispis rezultata predviđeno Edit3 polje. Neka su **x**, **y** i **z** realne promenljive. Pridruživanje vrednosti iz edit polja promenljivim se realizuje naredbama:

```
x:=StrToFloat(Edit1.Text);
```

```
y:=StrToFloat(Edit2.Text);
```

sabiranje se realizuje naredbom

```
z:=x + y;
```

Upis rezultata u Edit3 polje se realizuje naredbom

```
Edit3.Text:=FloatToStr(z);
```

(U Pascalu - Delphiju se ne pravi razlika između malih i velikih slova. Svejedno je da li ste otkucali *strtoint* ili *StrToInt*. Prilikom kucanja naredbi bide nuđena pomod [IntelliSense](#).)

Programiranje rukovaoca događajima

U odeljku **Komponenta Label** isprogramiran je događaj - ako se klikne levim tasterom miša na labelu menja se tekst labele u pozdravnu poruku **Dobar dan. Hvala sto ste kliknuli na mene.**

Događaji se pokredu akcijom korisnika. Na primer, kada korisnik klikne na dugme ili promeni tekst u tekst boks (edit polju), događaj biva generisan. Akcija korisnika, kao što je klik ili dupli klik na komponentu ili prevlačenje kursora preko komponente takođe generiše događaje. Ovi događaji se koriste za izvršavanje Delphi kôda povezanog sa njima.

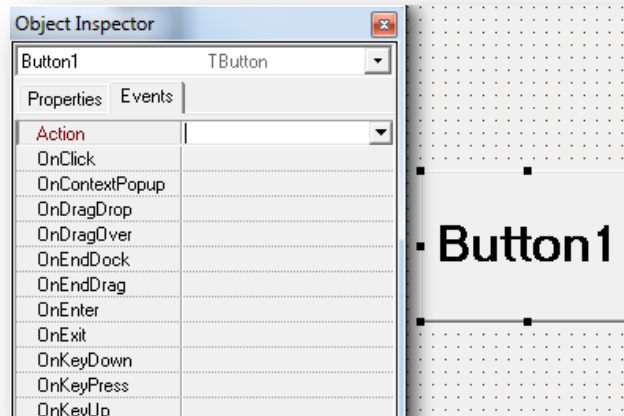
Kada smo kreirali ispis poruke, dva puta smo kliknuli na labelu i pojavio se prozor sa delom kôda za Label1.Click događaj. Između begin i end; ukucali smo naredbu

```
Label1.Caption:='Dobar dan. Hvala sto ste kliknuli na mene';
```

i kompletirali kôd događaja.








Za svaku komponentu postoji osnovni događaj. Kada se u dizajn (Design) modu, dva puta klikne na komponentu dobide se prozor sa kôdom za osnovni događaj. Na primer, za dugme je osnovni događaj Klik (OnClick), za tekst boks je osnovni događaj promena teksta (OnChange), za labelu je osnovni događaj klik (OnClick).

Kada se u Inspektoru objekata odabere kartica **Events** dobide se lista događaja koji se mogu vezati za odabranu komponentu. Na slededoj slici se vidi lista događaja koji se mogu dodeliti komponenti dugme (Button).



Slika 15

Neki od događaja sa liste se aktiviraju na sledeće načine:

-  *OnClick* - levi taster miša je pritisnut i otpušten;
-  *OnMouseDown* - levi taster miša je pritisnut;
-  *OnMouseMove* - kursor miša se pomera iznad komponente;
-  *OnMouseUp* - levi taster miša je otpušten;
-  *OnKeyDown* - pritisnut je taster na tastaturi;
-  *OnKeyPress* - taster na tastaturi je pritisnut i otpušten;
-  *OnKeyUp* - taster na tastaturi je otpušten;
- ...

Primer

Kreirati aplikaciju koja će ispisati pozdravnu poruku kada se klikne na dugme koje je dodato formi.

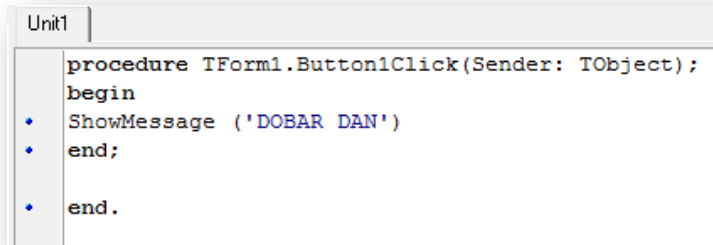
Dodademo dugme (Button) formi. Svojstvo Caption promenimo u **Klikni na mene**. U Inspektoru objekata odaberimo karticu **Events**. Odaberimo sa liste događaj **OnClick** tako što pomodu miša kliknemo dva puta u boks desno pored imena događaja. Pojavide se urednik koda u kome demo dopisati kod za akciju. Ovaj događaj se može vezati za dugme i brže - OnClick je osnovni događaj za komponentu dugme pa smo mogli samo kliknuti dva puta na dugme i pojavio bi se urednik koda sa događajem OnClick. Dok ne uvežbate dodavanje događaja komponentama, koristite karticu Design Inspektora objekata. Uvek proverite da li u nazivu procedure preko koje programiramo događaj stoji ime događaja koji ste želeli da vežete za komponentu. U nazivu procedure treba da piše:

`procedure TForm1.Button1Click`

ili, ako ste podešavali imena (svojstvo Name) forme i dugmeta, onda će u nazivu procedure pisati:

`procedure imeforme.imedugmetaClick`

U uredniku koda dokucajte tekst sa slike.

A screenshot of a Delphi IDE window titled 'Unit1'. The code editor shows a procedure definition for a button click event. The code is as follows:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
• ShowMessage ('DOBAR DAN')  
• end;  
  
• end.
```

Slika 16

i startujte program.

Procedura ShowMessage('Dobar dan') de u dijalog prozoru ispisati tekst sa porukom napisanom između apostrofa (Dobar dan). Često se u programu javlja potreba za slanjem poruka korisniku. Efikasan način jeste da se to uradi pomoću ove procedure koja u iskačudem dijalog prozoru ispiše poruku navedenu između apostrofa i čeka potvrdu korisnika za nastavak rada (klik na dugme **Ok** u dijalog prozoru).