

```
    Delay (0, 5);  
end;  
Panel1.Visible := True;
```

Da bih izbegao da skrolovanje bude prebrzo, naročito na bržim kompjuterima, ja sam unutar petlje `for` dodao poziv proceduri `Delay` koja kao parametre očekuje sekunde ili milisekunde pauze. Ova procedura `Delay` proverava trenutno vreme, a zatim čeka u petlji `while` sve dok ne prođe zahtevani broj sekundi ili milisekundi:

```
procedure Delay (Seconds, MilliSec: Word);  
var  
    TimeOut: TDateTime;  
begin  
    TimeOut := Now + EncodeTime (0,  
        Seconds div 60, Seconds mod 60, MilliSec);  
    // wait until the TimeOut time  
    while Now < TimeOut do  
        Application.ProcessMessages;  
end;
```

Unutar petlje sam pozvao metod `ProcessMessages` globalnog objekta `Application` da bih Windowsu omogućio da generiše i pošalje potrebnu poruku za iscrtavanje. Ova procedura `Delay` je zaista generička te je lako možete koristiti i u drugim aplikacijama.

#### SAVET

Uzmite u obzir i drugi aspekt prethodne procedure. Mi smo napisali kod kojim se nešto iscrtava na površini okvira za dijalog. Mada nije uobičajeno, okviri za dijalog mogu da imaju grafički izlaz i mogu da reaguju na akcije mišem baš kao i svaki drugi formular. Zapravo, okvir za dijalog i jeste formular. ■

## Izrada uvodnog ekrana

Još jedna tipična tehnika koja se koristi u aplikacijama je prikazivanje inicijalnog ekrana pre nego što se prikaže glavni formular. Ovo aplikaciju čini ugodnijom jer korisniku prikazujete nešto dok se program učitava, ali takođe predstavlja lep vizuelni efekat. Ponekad se isti prozor prikazuje i za okvir `About`.

Kao primer u kome je uvodni ekran naročito koristan, ja sam izradio program koji prikazuje listu sa prostim brojevima. Prosti brojevi se izračunavaju prilikom pokretanja programa tako da se prikazuju čim formular postane vidljiv:

```
procedure TForm1.FormCreate (Sender: TObject);  
var  
    I: Integer;  
begin  
    for I := 1 to 20000 do  
        if IsPrime (I) then  
            ListBox1.Items.Add (IntToStr (I));  
end;
```

Ovaj metod poziva funkciju `IsPrime` koju sam dodao programu. Ova funkcija, koju možete pronaći u izvornom kodu, izračunava proste brojeve veoma sporo; ali, ja sam morao da usporim

kreiranje formulara da bih Vam pokazao poentu. Brojevi se dodaju u listu koja pokriva celu klijent oblast formulara i omogućava da se prikaže više kolona, kao što možete videti na slici 8.8.

Kao što vidite prilikom pokretanja primera Splash0, problem je u tome što inicijalna operacija, koja se obavlja u metodu `FormCreate`, zahteva dosta vremena. Kada pokrenete program, potrebno je nekoliko sekundi (na standardnoj Pentium mašini) da se prikaže formular. Ukoliko je Vaš kompjuter veoma brz ili veoma spor, možete da promenite gornju granicu petlje `for` metoda `FormCreate` da biste program učinili bržim ili sporijim.

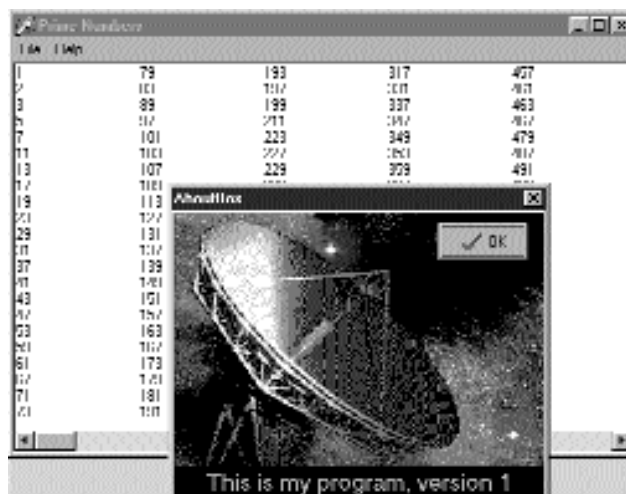
Ovaj program sadrži jednostavan okvir za dijalog sa slikom, jednostavno zaglavlje, bitmapiranu kontrolu, i svi su smešteni unutar panela i zuzimaju celu površinu okvira `About`. Ovaj formular se prikazuje kada odaberete element menija `Help`→`About`. Ali, ono što zaista želimo da prikazemo je okvir `About` prilikom pokretanja programa. Ovaj efekat možete da vidite pokretanjem primera `Splash1` i `Splash2`, koji prikazuju uvodni ekran na dva različita načina.

Pre svega, ja sam dodao metod klasi `TAboutBox`. Ovaj metod, nazvan `MakeSplash`, menja neka svojstva formulara da bi formular učinio pogodnim za uvodni ekran. U osnovi, metod uklanja borduru i zaglavlje, sakriva kontrolu `OK`, borduru panela čini tankom (da bi se zamenila bordura formulara) i zatim prikazuje formular momentalno ga ponovo iscrtavajući (videti sliku 8.9):

```

procedure TAboutBox.MakeSplash
begin
  BorderStyle := bsNone;
  BitBtn1.Visible := False;
  Panel1.BorderWidth := 3;
  Show;
  Update;
end;

```



**SLIKA 8.8** Glavni formular primera `Splash`, kada je okvir `About` aktiviran iz menija



**SLIKA 8.9** Formular uvodnog ekrana primera *Splash1* je malo drugačiji od originalnog okvira *About* (prikazanog na slici 8.8)

Ovaj metod se poziva posle kreiranja formulara u projekt-fajlu primera *Splash1*. Ovaj kod se izvršava pre kreiranja ostalih formulara (u ovom slučaju samo glavnog formulara), a zatim se uklanja uvodni ekran pre izvršavanja aplikacije. Ove operacije se obavljaju u bloku `try-finally`. Evo izvornog koda glavnog bloka projekt-fajla za primer *Splash2*:

```
var
    SplashAbout: TAboutBox;

begin
    Application.Initialize;
    // create and show the splash form
    SplashAbout := TAbouttox.Create (Application);
    try
        SplashAbout.MakeSplash;
        // standard code...
        Application.CreateForm(TForm1, Form1);
        // get rid of the splash form
        SplashAbout.Close;
    finally
        SplashAbout.Free;
    end;

    Application.Run;
end.
```

Ovaj pristup ima smisla samo ukoliko je za glavni formular Vaše aplikacije potrebno nešto vremena za kreiranje, za izvršavanje koda (kao što je to ovde slučaj), ili za otvaranje tabela baze podataka. Primetićete da je uvodni ekran prvi formular koji je kreiran, ali, pošto program ne koristi metod `FormCreate` objekta `Application`, ovaj formular ne postaje glavni formular aplikacije. U tom slučaju, zatvaranje uvodnog ekrana bi prekinulo izvršavanje programa!

Alternativni pristup je nešto duže zadržavanje uvodnog ekrana i upotreba tajmera da bi ga se oslobodili posle nekog vremena. Ja sam ovu drugu tehniku implementirao u primeru *Splash2*. Ovaj primer takođe koristi drugačiji pristup kreiranju uvodnog ekrana: umesto kreiranja uvodnog ekrana izvornim kodom, primer kreira formular na samom početku metoda `FormCreate` glavnog formulara.

```
procedure TForm1.FormCreate(Sender: TObject);
var
  I: Integer;
  SplashAbout: TAboutBox;
begin
  // create and show the splash form
  SplashAbout := TAboutBox.Create (Application);
  SplashAbout.MakeSplash;
  // standard code...
  for I := 1 to 20000 do
    if IsPrime (I) then
      ListBox1.Items.Add (IntToStr (I));
  // get rid of the splash form, after a while
  SplashAbout.Timer1.Enabled := True;
end;
```

#### NAPOMENA

Ovaj kod se pravilno izvršava bez obzira na redosled kreiranja formulara, kao što je naznačeno svojstvom `OldCreateOrder` (razmatrali smo ovo svojstvo u Poglavlju 6). ■

Tajmer se aktivira pre nego što se završi izvršavanje metoda. Kada protekne određeni period (u primeru je to tri sekunde), aktivira se događaj `OnTimer`, a uvodni ekran ga obrađuje tako što će se zatvoriti i ukloniti formular uvodnog ekrana:

```
procedure TAboutBox.Timer1Timer (Sender: TObject);
begin
  Close;
  Release;
end;
```

#### NAPOMENA

Metod `Release` formulara je sličan metodi `Free` objekata, jedino što se uklanjanje formulara odlaže sve dok se ne izvrše sve obrade događaja. Upotreba metoda `Free` može da dovede do narušavanja pristupa jer se interni kod, koji pokreće obradu događaja, može odnositi na formular. ■

Potrebno je popraviti još jednu stvar. Formular `Main` će biti kasnije prikazan ispred uvodnog ekrana, izuzev ukoliko ne učinite da bude na vrhu. Zbog toga sam dodao jednu liniju metodu `MakeSplash` okvira `About` u primeru `Splash2`:

```
FormStyle := fsStayOnTop;
```

## Formulari sa više strana

Kada je potrebno da prikazete dosta informacija i kontrola u okviru za dijalog ili formularu, tada možete da koristite više strana. To je metafora beležnice: upotrebom kartica korisnik može odabrati jednu od mogućih strana.

Postoje dve kontrole koje možete upotrebiti za izradu višestranične aplikacije u Delphiju: