

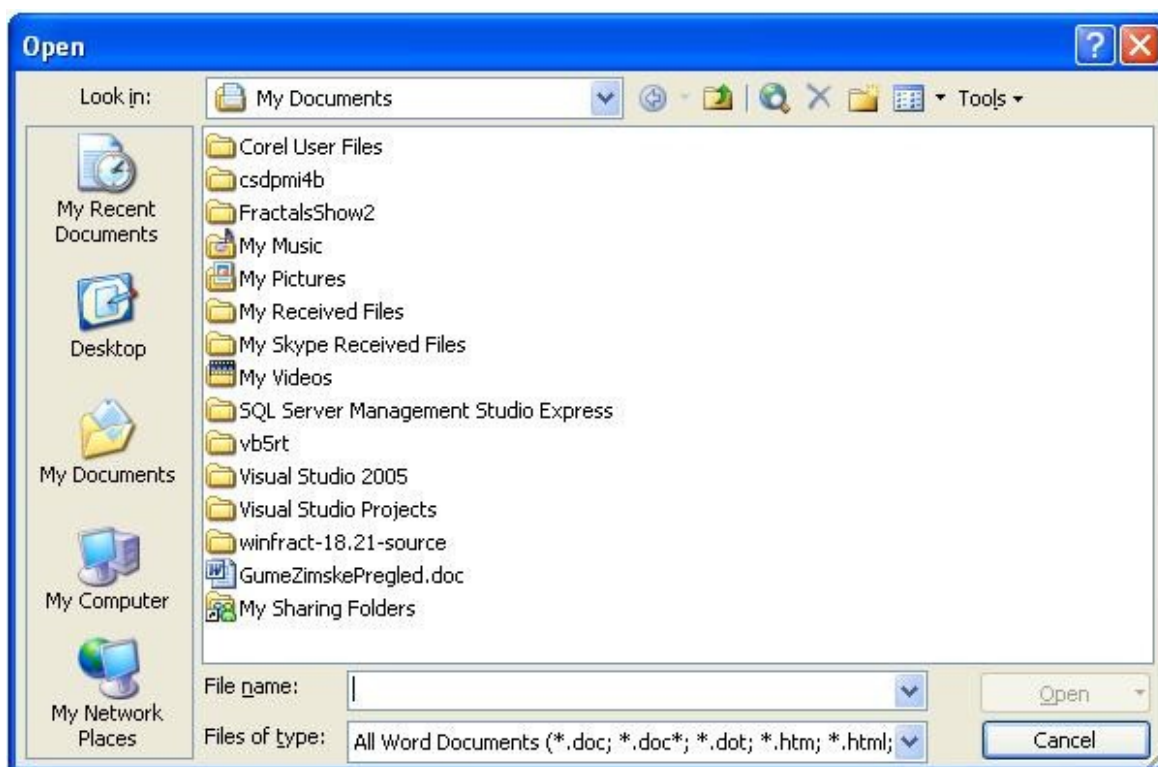


Slika 30: Rezultat postavljanja svojstvo Enabled na False za stavku „RadioGroup forma”

Rad sa standardnim dijalogima i datotekama

Većina Windows aplikacija ima potrebe da manipuliše različitim datotekama (fajlovima) (tekstualne datoteke, slike, itd.) u smislu čitanja i pisanja sadržaja.

Manipulacija datotekama obavezno uključuje i standardne dijaloge za izbor i čuvanje sadržaja. Oni se otvaraju izborom opcija **Snimi** (eng. Save) i **Otvori** (eng. Open). Na primer, u Microsoft Wordu 2003 dijalog za otvaranje dokumenta je prikazan na slici 31.



Slika 31: Open dijalog u aplikaciji Microsoft Word 2003

Bez obzira na vrstu dijaloga osnovne funkcionalnosti su iste: Moguće je izabrati datoteku, prikazane su samo datoteke koje zadovoljavaju određeni kriterijum (filter tj. tip datoteka) (All Word Documents na slici 31) uz mogućnost da se ručno unese ime datoteke - File name polje.

Ovaj dijalog je relativno složen i za njegovo programiranje treba uložiti dosta napora što se može zaključiti iz dosadašnjih lekcija. Upravo zbog toga su dijalozi, koji se često sreću u aplikacijama, deo mogućnosti razvojnog okruženja i nude se kao gotovi objekti. Znači nije ih potrebno ručno programirati! Oni su sistemski i postoje u svakoj verziji Windowsa. Čak, ako na primer kreirate aplikaciju u Windowsu XP sa ovim dijalozima i kopirate izvršnu datoteku na drugom OS npr. Windows Vistu, videćete Vistine dijaloge bez ikakve izmene aplikacije.

Korišćenje ovih dijaloga je veoma jednostavno, direktno i potpuno podržano u razvojnem okruženju VisualStudio. (u svim verzijama).

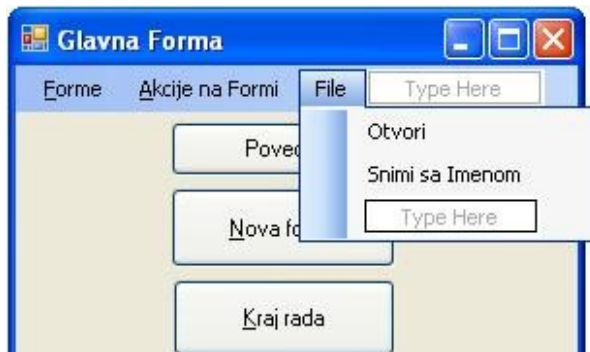
Korišćenje ovih dijaloga se može opisati sledećim koracima:

1. Postaviti odgovarajuću kontrolu na formu.
2. Postaviti potrebna svojstva kontrole (naslov, početni direktorijum, filter za ekstenzije datoteka, ...).
3. Otvaranje dijaloga

4. Po zatvaranju dijaloga proveriti da li je korisnik izabrao datoteku ili kliknuo na dugme Odustani (eng. Cancel)

5. Ako je izabrana datoteka, iz svojstva **FileName** čitamo putanju i ime datoteke i dalje radimo sa njom

Demonstriraćemo način na koji se koristi Open dijalog tako što ćemo u našu aplikaciju da dodamo meni "File" sa stavkama "Otvori" i "Snimi sa Imenom" (slika 32).



Slika 32: Meni "File"

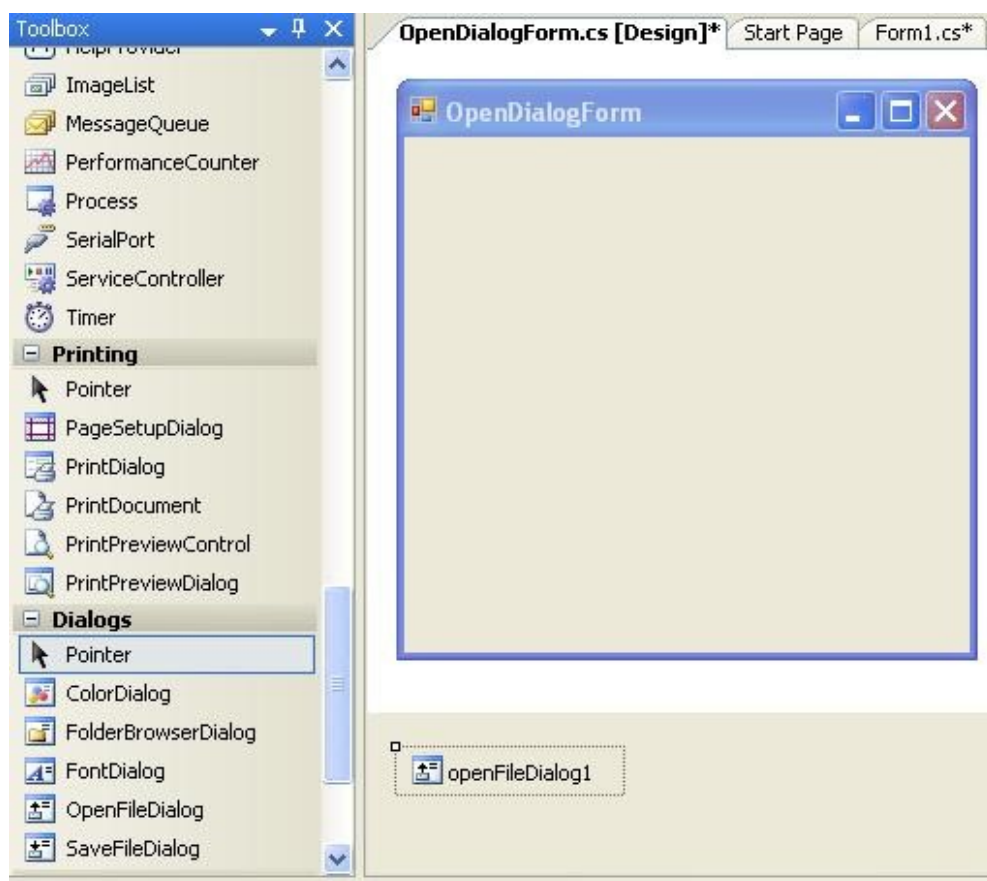
Nadalje kreirajmo novu formu „OpenDialogForm” na način kako je to ranije objašnjeno. Ova forma treba da se pojavi kada se aktivira stavka „Otvori” u meniju „File”. Da bi to postigli za događaj **Click** za stavku „Otvori” unesimo sledeći kod

```
private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    OpenFileDialog odf = new OpenFileDialog();
    odf.ShowDialog();
}
```

Dalje radimo u skladu sa prethodno opisanim koracima:

1. Postaviti odgovarajuću kontrolu na formu

U paleti sa alatima (sa leve strane razvojnog okruženja), pronađite sekciju "**Dialogs**" kao što je prikazano na slici 33. U ovoj sekciji se nalaze svi standardni Windows dijalozi. Osim pomenutih, tu su dijalozi za izbor boje, pregled direktorijuma i izbor fonta. Na formu „OpenDialogForm” postavite (prevuci i pusti (drag & drop) postupak) kontrolu **OpenFileDialog**. Kontrola se prikazuje na traci ispod forme i inicijalno je dobila naziv (svojstvo **Name**) **openFileDialog1**.



Slika 33: Dialogs sekcija u paleti sa alatima

2. Postaviti potrebna svojstva kontrole (naslov, početni direktorijum, filter za ekstenzije datoteka)

Kada pokrenemo dijalog za otvaranje datoteke, uobičajeno je da u njemu vidimo samo datoteke sa određenom ekstenzijom (tj. datoteke odgovarajućeg tipa) koju naša aplikacije prepoznaje. Obično se za svaki slučaj dodaje i opcija za prikaz svih datoteka - filter *.* sigurno ste do sada u ovakvim dijalogima videli opciju "All files" (sve datoteke).

Filter Open dijaloga se postavlja pomoću istoimenog svojstva **Filter**. Ovo svojstvo je tipa **string** i zahteva specifično formatiranje. Na primer, ako ovo svojstvo postavimo na:

Tekstualne datoteke|.txt|Sve datoteke|*.*

napravili smo dva filtra. Prvi, sa nazivom "**Tekstualne datoteke**", koji prikazuje sve datoteke sa ekstenzijom **txt**, i drugi sa nazivom "**Sve datoteke**", koji prikazuje upravo to: sve datoteke sa bilo kojom ekstenzijom.

Iz ovoga se vidi da se svaki filter sastoji od **dva** dela: naziva filtra i kriterijuma za prikaz datoteka. Separator je **vertikalna crta**. Obratite pažnju da na kraju ne treba staviti vertikalnu crtu.

Dozvoljene su i sledeće kombinacije:

`Slike*.jpg;*.gif;*.bmp|Video*.mpg;*.avi;*.wmv`

U ovom slučaju takođe imamo dva filtra, **Slike** i **Video** ali svaki od njih će prikazati datoteke sa više ekstenzija. Separator između ekstenzija istog filtra je karakter **tačkazarez** (;).

U našem primeru, svojstvo **Filter openFileDialog1** kontrole postavite na:

`Tekstualne datoteke*.txt|Sve datoteke|*.*`

Obično se postavlja i naslov dijaloga koji se definiše pomoću svojstva **Title**. Pošto u našem primeru treba otvoriti tekstualne datoteke postavite ovo svojstvo na **Otvori tekstualnu datoteku**.

Opciono možemo postaviti i naziv direktorijuma čiji će sadržaj biti prikazan odmah po otvaranju dijaloga. Ako želimo da se odmah prikaže koren diska C: u svojstvo **InitialDirectory** upišite **C:**.

Sada smo pripremili dijalog za otvaranje.

3. Otvaranje dijaloga

Dijalog poseduje metodu **ShowDialog** koja kao rezultat izvršavanja vraća akciju korisnika u dijalogu. Ova akcija predstavlja enumeraciju pod nazivom **DialogResult**. Nama su od interesa samo dve opcije:

- **DialogResult.OK** - korisnik je izabrao datoteku
- **DialogResult.Cancel** - korisnik je zatvorio dialog (kliknuo na Cancel dugme)

Proverom ove povratne vrednosti znamo da li je datoteka izabrana i dalje nastavljamo rad sa njom.

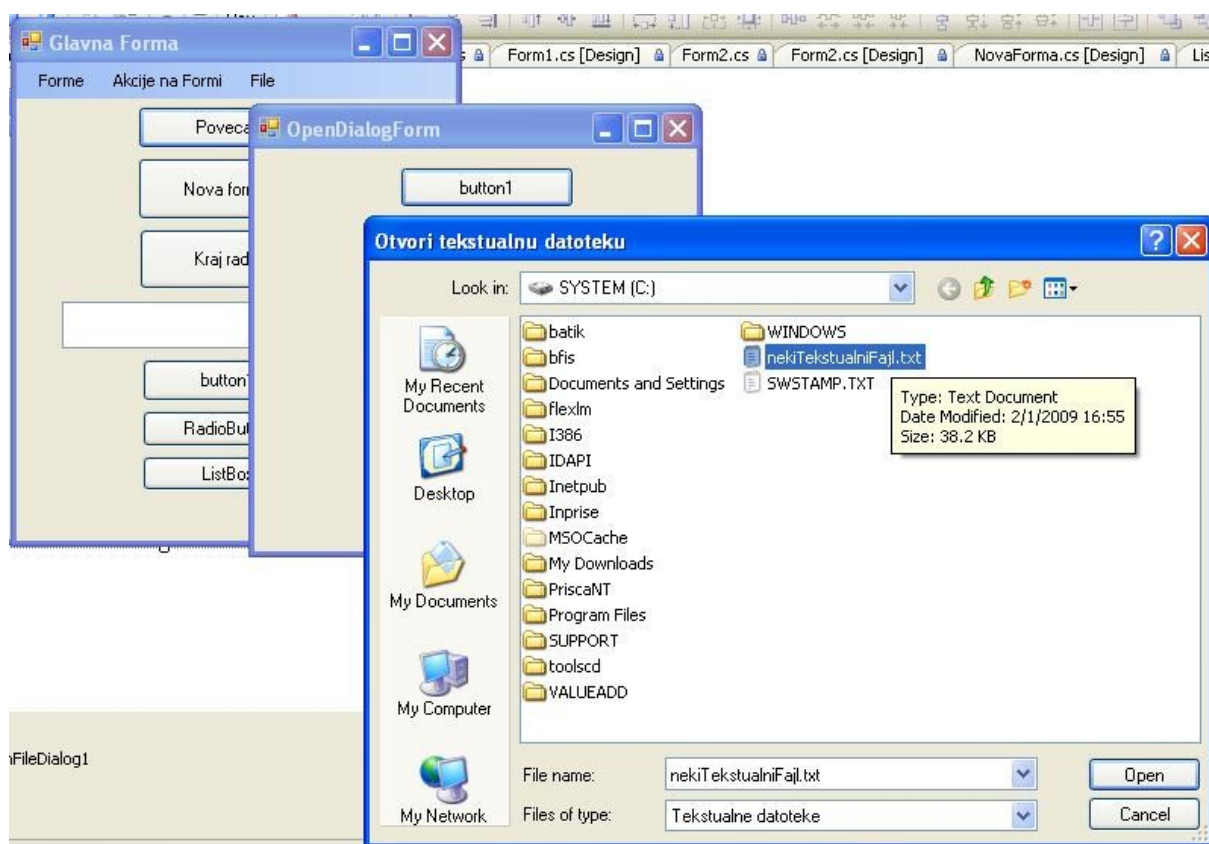
Na formu „OpenDialogForm” postavite jedno dugme i uradite dupli klik mišem na njega kako bi napisali sledeći kôd.

```
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        this.Text = openFileDialog1.FileName;
    }
}
```

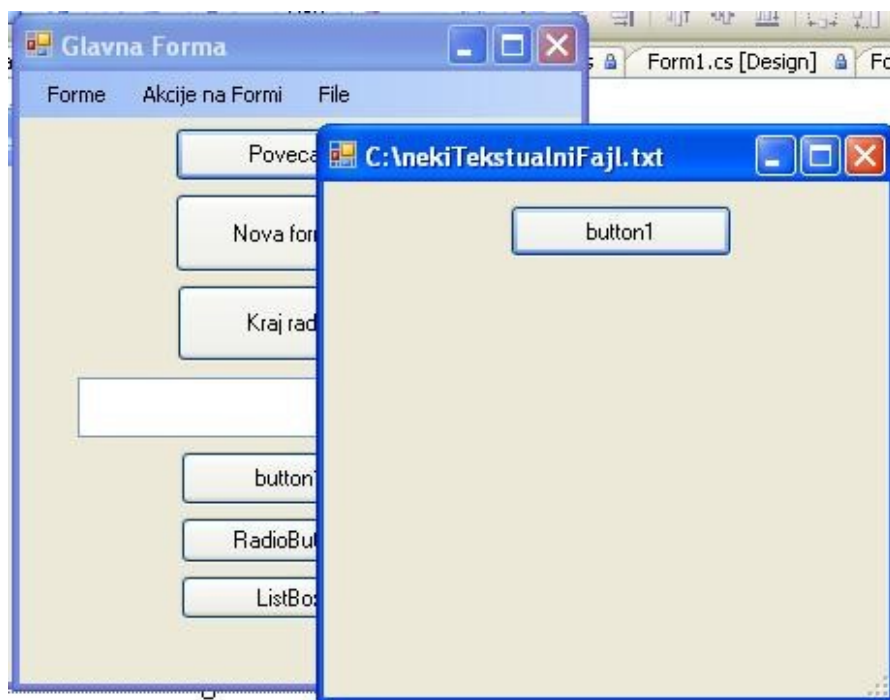
}

Pomoću metode **openFileDialog1.ShowDialog()** otvaramo dijalog i odmah proveravamo njegovu povratnu vrednost. Ako je ona jednaka **DialogResult.OK**, znači da je korisnik izabrao datoteku i kliknuo na dugme **“Open”** u dijalogu. U tom slučaju u svojstvu **FileName** se nalazi ime izabrane datoteke sa kompletnom putanjom. U naslovu formulara (**this.Text**) upisujemo putanju i ime izabrane datoteke. Dakle svojstvo **FileName** je ključno, jer definiše lokaciju izabrane datoteke u file sistemu.

Pokrenite program, aktivirajte stavku Otvori u meniju File a zatim kliknuti na dugme **Open**. Kao rezultat pojaviće se standardni dijalog za otvaranje fajla (slika 34). Obratite pažnju na dva filtra koja smo kreirali i proverite funkcionisanje oba. Na svom računaru pronađite odgovarajuću datoteku (u dijalogu prikazanom na slici 34 izabran je fajl nekiTekstualniFajl.txt) i kliknite na dugme **“Open”** na dijalogu. Nakon ovoga u naslovu forme će biti ispisana kompletna putanja sa nazivom izabrane datoteke (u našem slučaju „C: \nekiTekstualniFajl.txt“, vidi sliku 35.).



Slika 34: Standardni File Open dijalog.



Slika 35: Rezultat izbora fajla "nekiTekstualniFajl.txt".

Prikaz sadržaja datoteke

Otvaranje dijaloga i izbor datoteke, naravno ne otvara niti prikazuje njen sadržaj. Dijalogom smo samo selektovali neku postojeću datotetu. Sledeći logičan korak je da zaista i prikažemo sadržaj tekstualne datoteke. Za otvaranje, čitanje i pisanje po datotekama je zadužen skup klasa u imenovanom prostoru (eng. NameSpace) **System.IO**, te je neophodno da ga uključimo (`using System.IO;`) da bi mogli da ga koristimo.

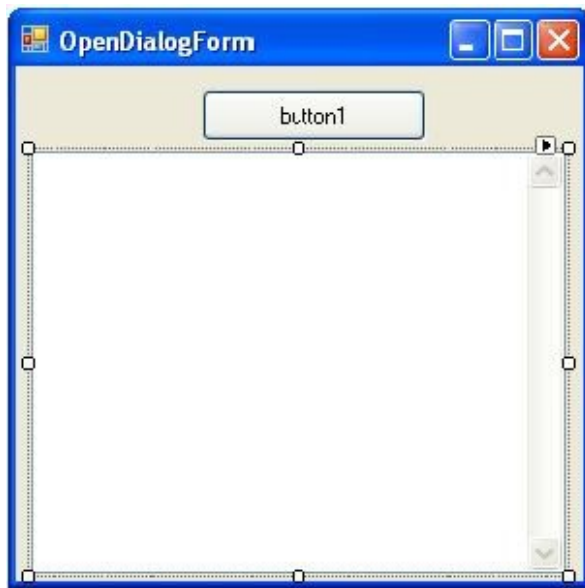
Za čitanje sadržaja datoteka koristi se klasa **StreamReader** kojoj se u konstruktoru zadaje putanja do konkretne datoteke. Putanju kao što smo videli već imamo u **FileName** svojstvu dijaloga. Metodom **ReadToEnd** StreamReader objekta čita se kompletan sadržaj zadate datoteke, i ova metoda vraća tip **string** koji ćemo prikazati u kontroli **TextBox**.

Zaustavite aplikaciju i na formu postavite **TextBox** kontrolu u kojoj će biti ispisan sadržaj tekstualne datoteke. Kontrola je inicijalno dobila ime (**Name** svojstvo) `textBox1`. Postavite joj sledeća svojstva na prikazane vrednosti:

Multiline: True

ScrollBars: Vertical

Forma "OpenDialog Form" sada treba da izgleda kao na slici 36.



Slika 36: Forma

Uradite dupli klik na dugme, kako bi modifikovali kôd na sledeći način:

```
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        this.Text = openFileDialog1.FileName;

        System.IO.StreamReader sr = new StreamReader(openFileDialog1.FileName);

        textBox1.Text = sr.ReadToEnd();

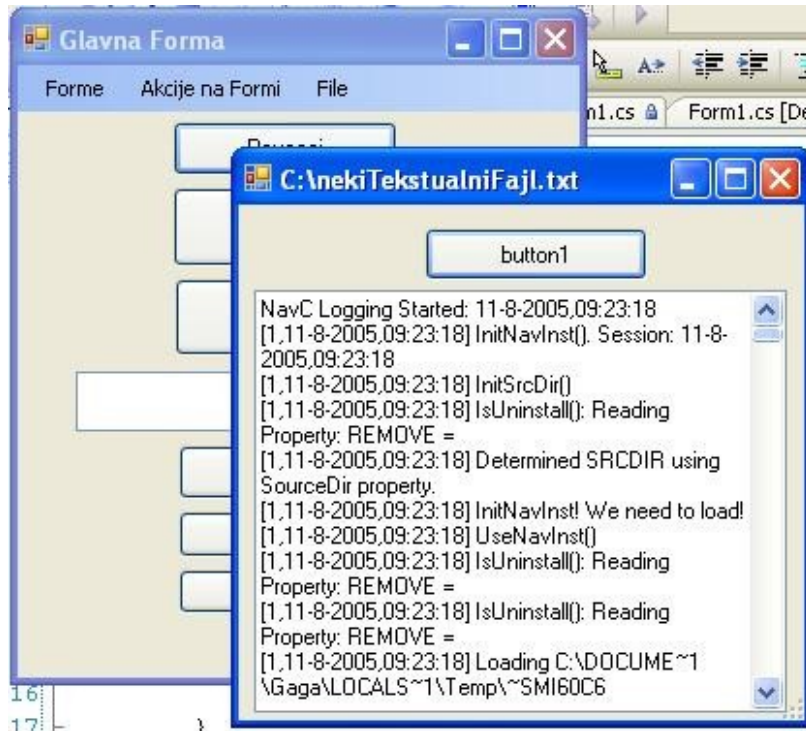
        sr.Close();
    }
}
```

Analizirajmo napisani kod. Inicijalizujemo objekat **sr** tipa `System.IO.StreamReader` i u konstruktoru postavljamo ime datoteke koju je korisnik izabrao u dijalogu. Metodom **ReadToEnd()** čitamo kompletni sadržaj datoteke i postavljamo ga u **Text** svojstvo kontrole **textBox1**.

Na kraju treba obavezno zatvoriti `StreamReader` objekat, što se radi metodom **Close()**. Ako startujemo aplikaciju i u dijalogu izaberemo tekstualni fajl njegov sadržaj će biti prikazan u `TextBoxu` kao što je to prikazano na slici 37.

UPOZORENJE:

Ako **StreamReader** ne zatvorite metodom **Close()**, datoteka koja se čita ostaje **zaključana** (eng. Locked) tako da ni jedna druga aplikacija ne može da je otvori. Kaže se da **StreamReader** datoteku otvara u **ekskluzivnom** režimu. Međutim, ako regularno zatvorite svoju aplikaciju, automatski se “otključavaju” sve otvorene datoteke. Znači obavezno je zatvaranje streamova!



Slika 37: Prikazan sadržaj datoteke nekiTekstualniFajl.txt.

Klasa Graphics

Od velikog broja aplikacija se zahteva da pored teksta i brojeva imaju i odgovarajući grafički prikaz. Klasa koja omogućuje crtanje po bilo kojoj sistemskoj kontroli je klasa **Graphics**. Objekat ove klase predstavlja površinu po kojoj se crta. Znači prvi korak je kreiranje ovog objekta a zatim njegovo korišćenje u skladu sa slikom koju želimo da nacrtamo. Objekat klase Graphics za neku kontrolu se kreira pozivom metode **CreateGraphics()** tako da je sada površina za crtanje ta kontrola. Klasa Graphics omogućuje crtanje različitih oblika kao što su: *linija*, *pravougaonik*, *popunjeni pravougaonik*, *elipsa*, *popunjena elipsa*, *kružni luk*, *kriva linija na osnovu 4 tačke*, itd. Metode koje se najčešće koriste za ovakva crtanja su:

- **DrawLine** – crtanje prave linije,