



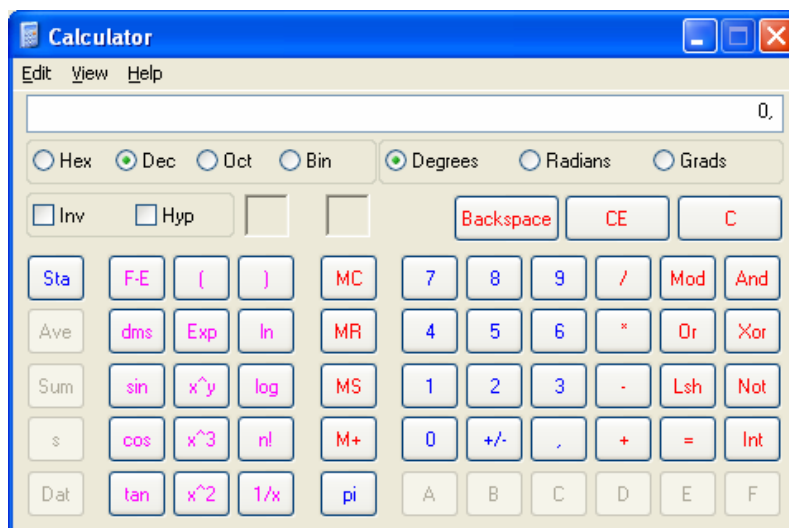
8

Elementi grafičkog korisničkog sučelja

- O grafičkom korisničkom sučelju
- Prozor i njegovi dijelovi
- Okviri za tekst i gumbi

O grafičkom korisničkom sučelju

Posljednji program koji smo napisali pokretali smo iz komandne linije operativnog sustava DOS. Parametre smo programu prosljeđivali tako da smo ih napisali neposredno iza imena programa. Današnji programi se najčešće ne pokreću iz komandne linije, a način komunikacije čovjeka s programom je pomoću mnoštva različitih elemenata (gumbi, okviri za tekst, padajuće liste, izbornici,...).



U ovom poglavlju ćemo naučiti kako kreirati program koji će s korisnikom komunicirati preko grafičkog korisničkog sučelja.

Osnovni element svakog takvog programa bit će **prozor**. Na njega ćemo doslovno lijepiti različite elemente, kao što su: okviri za tekst, gumbi, izbornici, labele, gumbi za odabir,...

Svaki od tih elemenata, pa i sam prozor su u stvari klase. I pojedini element ćemo dodavati tako da napravimo instancu odgovarajuće klase te mu definiramo neka svojstva.

Kao što znamo, na neke radnje unutar takvog korisničkog sučelja se događaju neke akcije. Npr. klikom na gumb 7 u programu kalkulator će se na displayu kalkulatora pojaviti broj 7. Općenito se takva reakcija zove **događaj** i svaki događaj unutar programa treba isprogramirati. Točno trebamo reći koji će biti odgovor na neku radnju.

Prozor i njegovi dijelovi

Već smo rekli da su prozor i općenito svi elementi grafičkog korisničkog sučelja u stvari klase. Te klase nisu inicijalno uključene u Javi, već ih moramo posebno pozvati. Klase za definiranje elemenata grafičkog korisničkog sučelja smještene su u paketu koji se naziva **javax.swing**. Općenito ćemo paket u klasu uključiti naredbom:

```
import ime_paketa.*;
```

Dakle, paket koji sadrži klase za generiranje elemenata grafičkog korisničkog sučelja pozvat ćemo naredbom:

```
import javax.swing.*;
```

znak `*` na kraju znači da uključujemo sve pakete koji se nalaze u paketu **javax.swing**.

Klasa koja će u Javi kreirati prozor je **JFrame**. Kao i svaka druga klasa i ova ima svoja svojstva i metode. Sama po sebi se nameće potreba da prozoru definiramo veličinu (širinu i visinu), naslov,... Pa upravo su te metode i definirane nad klasom **JFrame**.

Klasa **JFrame** ima dva konstruktora:

JFrame () – kreira prozor bez ikakvih elemenata


JFrame (String s) – kreira prozor u čijoj će naslovnoj traci biti pisati string koji smo proslijedili kao parametar.

Najčešće metode koje ćemo mi koristiti nad prozorom su:

public void setSize (int s, int v) – definira širinu i visinu prozora

public void setTitle (String s) – definira naslov prozora

public void setVisible (boolean b) – postavlja prozor u programu. Ukoliko je vrijednost varijable *b* postavljena na *true*, prozor će biti vidljiv.

public void setDefaultCloseOperation (int n) – definira akciju koja se događa klikom na gumb close (). Mi ćemo uvijek kao parametar ove metode pisati svojstvo *EXIT_ON_CLOSE* klase **JFrame**.

public Container getContentPane () – vraća središnji dio prozora (*Container*), na koji ćemo dodavati elemente grafičkog korisničkog sučelja (gumbe,...).

Primjer 8 – 1:

Napišimo program koji će na ekranu crtati prozor u čijoj će naslovnoj traci pisati "*Jedan običan prozor*", dok će mu širina biti 400 a visina 300 piksela.

Rješenje:

Budući da trebamo napisati program koji će crtati prozor na ekranu, u konstruktoru ćemo napraviti prozor, definirati mu sve elemente i postaviti ga vidljivim, a onda ćemo u metodi **main** kreirati instancu klase koja će (zbog definicije konstruktora) iscrtati prozor na ekranu.

```
import javax.swing.*;
public class Prozor
```

```

{
    private JFrame p = new JFrame ();
    public Prozor()
    {
        p.setTitle ("Jedan običan prozor");
        p.setSize (400, 300);
        p.setDefaultCloseOperation (p.EXIT_ON_CLOSE);
        p.setVisible (true);
    }

    public static void main (String[] s)
    {
        Prozor pr = new Prozor ();
    }
}

```

Pokretanjem metode **main** na ekranu će se pojaviti prozor kao na slici:



Prozor isto tako možemo kreirati i tako da kreiramo klasu koja nasljeđuje klasu **JFrame**. U tom slučaju nećemo morati kreirati instancu klase **JFrame**, što će nam omogućiti direktno pozivanje svojstava i metoda klase **JFrame**. U tom slučaju će rješenje prošlog primjera imati sljedeći oblik:

```

import javax.swing.*;
public class Prozor extends JFrame
{
    public Prozor()
    {
        setTitle ("Jedan običan prozor");
        setSize (400, 300);
        setDefaultCloseOperation (EXIT_ON_CLOSE);
        setVisible (true);
    }

    public static void main (String[] s)
    {
        Prozor pr = new Prozor ();
    }
}

```

Mi ćemo u daljnjim primjerima uglavnom kreirati prozore nasljeđivanjem klase **JFrame**.

Okviri za tekst i gumbi

U prošlom primjeru smo naučili kako kreirati jedan običan prozor. Nadalje ćemo prozorima dodavati dodatne komponente. Već smo rekli da su elementi grafičkog korisničkog sučelja u stvari klase.

Okviri za tekst

Okvir za tekst kreiramo klasom **JTextField**.

Instancu klase **JTextField** najčešće ćemo kreirati koristeći konstruktor bez parametara, a sva potrebna svojstva definirat ćemo koristeći metode. Neke od najčešće korištenih metoda u za rad s okvirom za tekst su:

```
public void setSize (int s, int v) – definira veličinu okvira za tekst
public void setLocation (int x, int y) – definira mjesto, unutar Containera, na kojem
će se nalaziti okvir za tekst (koordinate gornjeg lijevog vrha okvira za tekst)
public void setText (String s) – postavlja tekst u okvir za tekst
public String getText () – vraća tekst koji je upisan u okvir za tekst
public void setEditable (boolean b) – ukoliko je b postavljen na true tekst unutar okvira
za tekst će se moći mijenjati. Ukoliko je b postavljen na false sadržaj okvira za tekst neće biti
moguće mijenjati.
```

Gornja klasa omogućavala nam je kreiranje okvira za unos teksta. Osim što se može unositi tekst se isto tako može i ispisivati na prozoru. Na prozoru ćemo tekst ispisivati unutar posebnih elemenata grafičkog korisničkog sučelja, tzv. labela. Klasa koja će nam kreirati takve labela je **JLabel**.

Instancu klase **JLabel** ćemo isto tako najčešće kreirati bez konstruktora. Neke od metoda ove klase su:

```
public void setSize (int s, int v) – definira veličinu labela
public void setLocation (int x, int y) – definira mjesto, unutar Containera, na kojem
će se nalaziti labela (koordinate gornjeg lijevog vrha okvira za tekst)
public void setText (String s) – postavlja tekst u labelu
```

Gumbi

Gumb ćemo kreirati klasom **JButton**.

Slično kao i kod okvira za tekst, instancu klase **JButton** ćemo uglavnom kreirati koristeći konstruktor bez parametara, a dodatna svojstva ćemo definirati pomoću metoda. Metode koje ćemo najčešće koristiti s gumbima su:

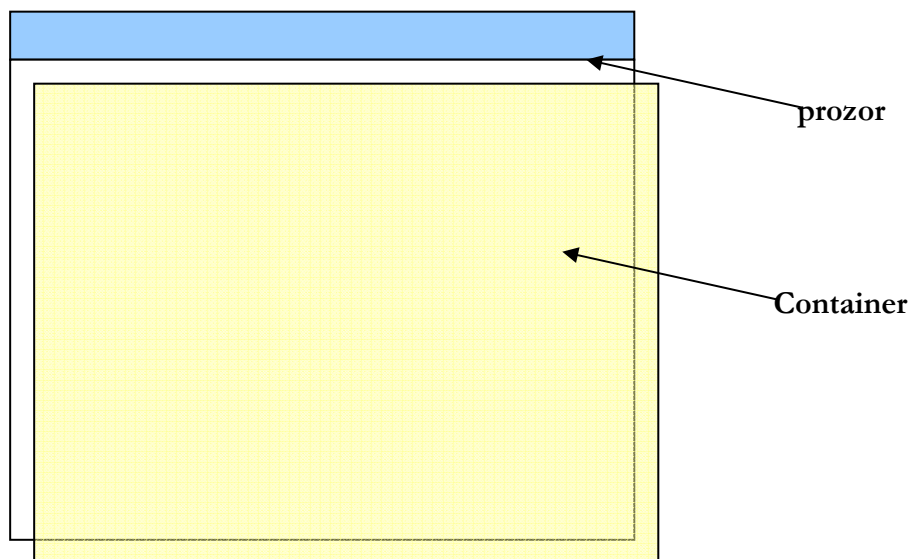
```
public void setSize (int s, int v) – definira veličinu gumba
public void setLocation (int x, int y) – definira mjesto, unutar Containera, na kojem
će se nalaziti gumb (koordinate gornjeg lijevog vrha gumba)
public void setText (String s) – postavlja tekst na gumb
public void setEnabled (boolean b) – ukoliko je vrijednost varijable b true gumb će biti
aktivan, inače gumb neće biti aktivan.
```

Elemente kao što su gumbi ili okviri za tekst ne možemo dodavati direktno na prozor koji smo kreirali u prethodnom primjeru. Komponente ćemo dodavati na jedan poseban dio prozora koji nazivamo *Container*. *Container* je opet jedna posebna klasa. Slično kao i klasa **JFrame**, i ova klasa se nalazi u jednom posebnom paketu: **java.awt**.

Slično kao i kod paketa **javax.swing**, ovaj paket ćemo učitavati naredbom:

```
import java.awt.*;
```

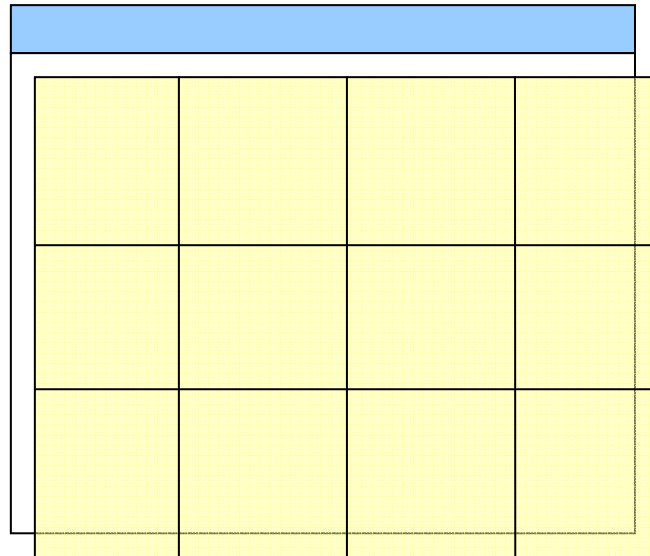
Instanci klase **Container** ćemo pridružiti središnji dio prozora metodom **getContentPane ()**.



Prije nego što počnemo dodavati elemente na *Container* moramo definirati izgled *Container*a. Izgled *Container*a ćemo definirati metodom **setLayout (LayoutManager m)** koja je definirana u klasi *Container*. Ovdje se nećemo upuštati u objašnjavanje **LayoutManagera**. U osnovi se radi o obliku *Container*a. Npr. područje *Container*a može biti tablica i elemente jednostavno redom dodajemo u ćelije tablice. Prvi element dodajemo u prvu ćeliju prvog reda, sljedeći u drugu ćeliju prvog retka,... posljednji element dodajemo u posljednju (desnu) ćeliju posljednjeg retka tablice.

Područje *Container*a koje će imati oblik tablice od 3 retka i 4 stupca definirat ćemo na sljedeći način:

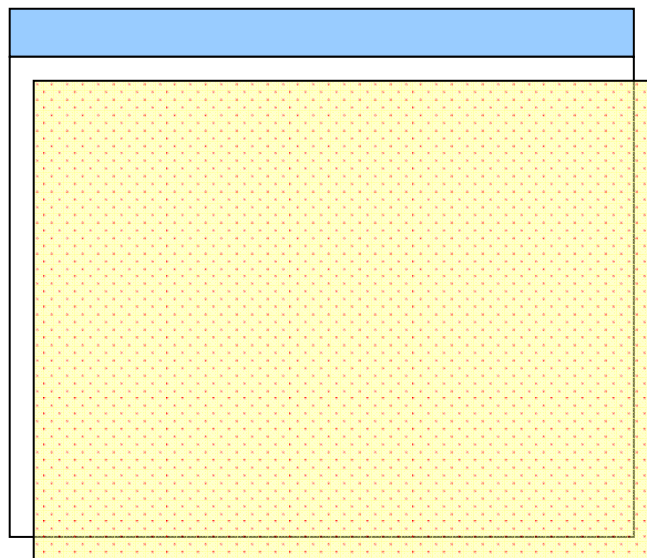
c.setLayout (new GridLayout (3, 4)), pri čemu je *c* instanca klase *Container*.



Mi ćemo ponekad elemente stavljati na određene koordinate *Containera*, jer će nam tako biti lakše. U tom ćemo slučaju područje *Containera* definirati na sljedeći način:

```
c.setLayout (null);
```

Ovime smo u stvari rekli da je naše područje prazna ploča i za svaki element ćemo reći na koje koordinate te ploče dolazi.



I na kraju ćemo još samo naučiti kako gotovi element dodati na *Container*. To ćemo napraviti metodom **add** klase *Container*. Opći oblik klase **add** je:

```
public void add (Object o);
```

Primjer 8 – 2:

Napišimo program koji će na ekranu nacrtati prozor. Prozor treba imati dva okvira za tekst, jedan ispod drugoga, ispod njih treba imati gumb na kojem piše *Zbroji* te ispod gumba još jedan okvir za tekst.

Rješenje:

```
import javax.swing.*;
import java.awt.*;
public class Zbrajanje extends JFrame
{
    private final int sirina = 220;
    private final int visina = 200;
    private Container c;
    private JTextField t1, t2, t3;
    private JButton b;

    public Zbrajanje()
    {
        setTitle ("Zbrajanje");
        setSize (sirina, visina);
        setDefaultCloseOperation (EXIT_ON_CLOSE);

        c = getContentPane ();
        c.setLayout (null);

        //prvi okvir za tekst
        t1 = new JTextField ();
        t1.setSize (200, 25);
        t1.setLocation (5, 5);
        c.add (t1);

        //drugi okvir za tekst
        t2 = new JTextField ();
        t2.setSize (200, 25);
        t2.setLocation (5, 45);
        c.add (t2);

        //gumb
        b = new JButton ();
        b.setText ("Zbroji");
        b.setSize (200, 25);
        b.setLocation (5, 85);
        c.add (b);

        //treći okvir za tekst
        t3 = new JTextField ();
        t3.setSize (200, 25);
        t3.setLocation (5, 125);
        c.add (t3);

        setVisible (true);
    }

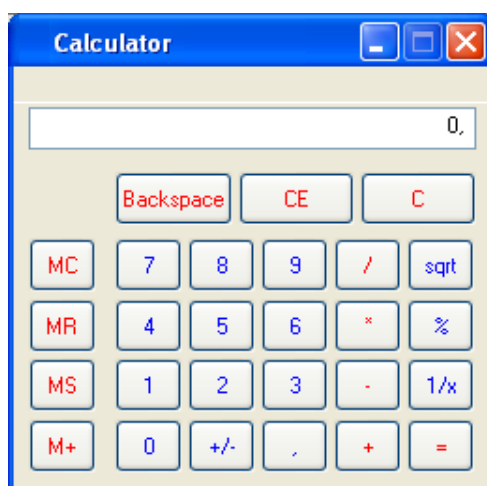
    public static void main (String[] s)
    {
        Zbrajanje z = new Zbrajanje ();
    }
}
```

Pokretanjem metode **main**, na ekranu će se pojaviti sljedeći prozor:



Zadaci za vježbu

1. Što je grafičko korisnikčko sučelje?
2. Koji paket moramo uključiti u program da bismo mogli raditi s prozorima?
3. Koja klasa omogućava crtanje prozora na ekranu?
4. Što je container?
5. Koja klasa omogućava kreiranje gumba na prozoru?
6. Koja klasa omogućava kreiranje okvira za tekst na prozoru?
7. Definiraj prozor koji će oko sredine imati jedan gumb na kojem će pisati **OK**.
8. Definiraj prozor koji će imati sljedeći oblik (nije potrebno voditi računa oboji teksta na gumbima):





9

Događaji

- Događaji