PT&SCHE

Introduction of part-time and short cycle studies in Serbia

Funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

UNIVERZITET METROPOLITAN BEOGRAD

KI105 - JAVA 3: PROGRAMIRANJE KORISNIČKOG INTERFEJSA

Osnove grafičkog korisničkog interfejsa u Javi (GUI)

Lekcija 01

PRIRUČNIK ZA STUDENTE

KI105 - JAVA 3: PROGRAMIRANJE KORISNIČKOG INTERFEJSA

Lekcija 01

OSNOVE GRAFIČKOG KORISNIČKOG INTERFEJSA U JAVI (GUI)

- Osnove grafičkog korisničkog interfejsa u Javi (GUI)
- Poglavlje 1: Swing Java klase
- Poglavlje 2: Okviri (JFrame)
- Poglavlje 3: Menadžeri rasporeda
- Poglavlje 4: Upotreba panela kao podkontejnera
- Poglavlje 5: Klase Color i Font
- Poglavlje 6: Zajednička svojstva Swing GUI komponenata
- Poglavlje 7: Ikone slika
- Poglavlje 8: Klasa JButton
- Poglavlje 9: Klasa JCheckBox
- Poglavlje 10: Klasa JRadioButton
- Poglavlje 11: Natpisi (JLabel)
- Poglavlje 12: Polje teksta (JTextField)
- Poglavlje 13: Domaći zadaci
- Zaključak

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

Uvod

UVOD

Cilj lekcije je da nauči studenta kako da kreira jednostavni grafički korisnički interfejs (GUI)

Cilj ove lekcije je da student nauči da:

- Razlikuje Swing i AWT pakete Java klasa
- Opiše Java GUI API
- Kreira korisnički interfejs upotrebom okvira, panela i jednostavnih GUI komponenti
- Razume ulogu menadžera rasporeda i upotrebu **FlowLayout, GridLayout i BorderLayout klasa koje raspoređuju komponente u kontejneru**
- Upotrebi **JPanel za grupisanje komponenti u podkontejner**
- Kreira objekte u boji uz pomoć **Color klase**
- Kreira objekte za slovne oznake upotrebom klase Font
- Primeni zajednička svojstva kao što su ivice, fontove, i boje kod Swing komponenata
- Dekoriše ivice GUI komponenti
- Kreira ikone sa slikama upotrebom **ImageIcon klase**
- Kreira i upotrebe dugmad upotrebom **JButton klase**
- Kreira i upotrebe **JCheckBox komponentu**
- Kreira i upotrebe **JRadioButton box komponentu**
- Kreira i upotrebe **JLabel komponentu**
- Kreira i upotrebe **JTextField komponentu**

Poglavlje 1

Swing Java klase

ABSTRACT WINDOWS TOOLKIT (AWT) I SWING

Swing komponente su „lake“ komponente, za razliku od AWT komponenata koje se „teške“ jer zavise mnogo više od GUI resursa i specifične platforme na kojoj se primenjuju

Kada je nastala Java, njeni autori su ponudili paket klasa pod nazivom Abstract Windows Toolkit (AWT) koja je omogućila razvoj jednostavnih grafičkih korisničkih interfejsa za komunikaciju korisnika nekog programa i samog programa. Ispostavilo se da AWT nije pogodan paket za složenije zahteve i da ima bagove pri primeni na pojedinim platformama. Zato su autori Jave razvili novi paket koji omogućava izradu robusnijih, pouzdanijih i složenih grafičkih komponenata za komunikaciju korisnika sa nekom aplikacijom, pod nazivom Swing. Swing komponente se direktno boje u Java kodu, sem komponenata koje su podklase klasa java.awt.Window i java.awt.Panel, koji se koristi GUI specifične platforme.

Swing komponente manje zavise od ciljne platforme i koriste manje specifične GUI resurse platforme na kojoj se primenjuju. Kako ne zavise od specifičnih GUI resursa platforma, Swing komponente se često nazivaju „lakim“ komponentama, za razliku od AWT komponentama koje se nazivaju „teškim“ komponentama, jer zavise mnogo više od GUI resursa specifične platforme na kojoj se primenjuju.

Da bi se se istakla razlika Swing komponenta od AWT komponenti, autori Jave su Swing komponentama dali nazive koji uvek počinju sa slovom J. Programerima se preporučuje da koriste Swing, a ne AWT komponente.

JAVA GUI API

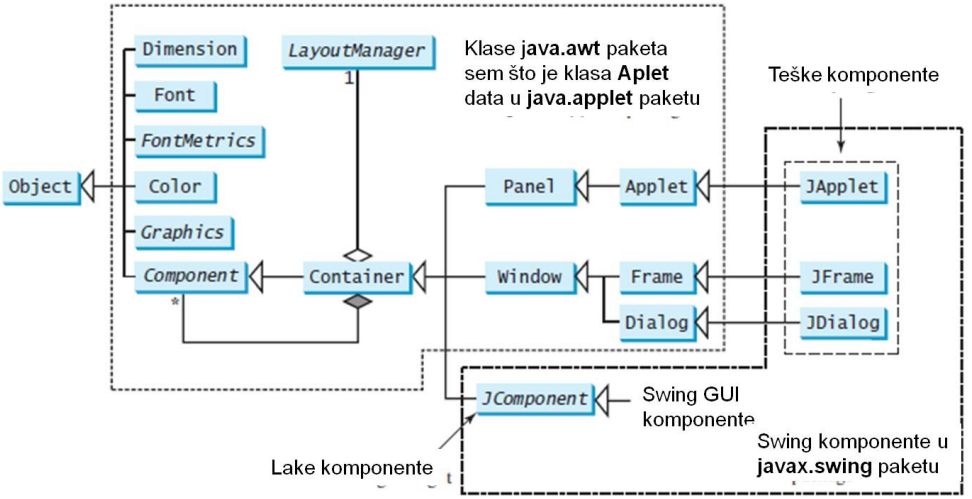
Java GUI API je hijerarhijska struktura klasa Swing paketa.

Na slici 1 prikazana je hijerarhijska struktura klasa ***Swing*** paketa Java GUI API.

Klasa ***Container*** je podklasa klase ***Component,*** koja ima tri podklase: ***Panel, Window i JComponent.*** Podklase klase ***Window*** su ***JFrame i JDialog*** koje prikazuju okvir u okviru koga se prikazuju ostale GUI komponente. Klasa***Panel***, odn. njena podklasa ***JApplet*** predstavlja ploču (koja se smešta u okvir) na kojoj se prikazuju komponente klase ***JComponent***. Klasa ***JComponent*** ima više podklasa za prikazivanje različitih komponenti. Kao što se vidi na slici 1, jedan objekat tipa ***Component*** može da sadrži više objekata tipa ***Container,*** različitog tipa (***Panel, Window i JComponen***t).

Klase, kao što su Graphics, Color, Font, FontMetrics i Dimension su tzv. pomoćne klase koje podržavaju navedene GUI komponente.

Klase JFrame, JApplet, JDialog i JComponent i njihove podklase su deo paketa javax.swing. Klasa Applet je deo paketa java.applet. Ostale klase na slici 1 su deo paketa java.awt.



Slika 1.1 Hijerarhijska struktura Java GUI

PRIMER 1 - KLASE KOMPONENATA (CONTAINER I JCONTAINER)

Klasa JComponent je koren za sve lake Swing komponente

Objekti klase ***Component*** se prikazuju na monitoru. Klasa ***Component*** je koren svih klasa korisničkog interfejsa, uključujući klase tipa kontejner. Klasa***JComponent*** je koren za sve lake Swing komponente. Klase ***Component*** i ***JComponent*** su apstrakte klase jer nemaju svoje posebne objekte (primerke klase). Za kreiranje određenih objekata, moraju se koristit podklase ovih klasa. Na primer, može se kreirati objekat klase JButton, i prikazati klase koje za koje je klasa***JButton*** podklasa (što se i vidi na slici 1):

```
JButton jbtnOK = new JButton("OK");
System.out.println(jbtnOK instanceof JButton);
System.out.println(jbtnOK instanceof JComponent);
System.out.println(jbtnOK instanceof Container);
System.out.println(jbtnOK instanceof Component);
System.out.println(jbtnOK instanceof Object);
```

KONTEJNER KLASE (CONTAINER)

Da bi radili samo sa Swing komponentama, koristite klase Container, JFrame, JDialog, JApplet, i JPane

Objekat (primerak) klase Container je kontejner najvišeg nivoa, ako nije ugrađen u neki drugi kontejner. Klase ***Window, Frame, Dialog, JFrame, i JDialog*** kontejneri najvišeg nivoa. Klase ***Window, Panel, Applet, Frame, i Dialog*** su kontejner klase koje mogu da sadrže ***AWT*** komponente. Da bi radili samo sa Swing komponentama (što se preporučuje), koristite klase ***Container, JFrame, JDialog, JApplet, i JPanel*** kao što je objašnjeno u Tabeli 1 .

Tabela 1: Kontejner klase

Klasa Container	Opis
`java.awt.Container`	Sadrži komponente. Njegove podklase su okviri, paneli i apleti.
`java.swing.JFrame`	Je kontejner na naivišem nivou i sadrži Swing komponente JavaGUI
`java.swing.JPanel`	Nevidljiv kontejner za grupisanje GUI komponenata. Paneli se mogu povezivati unutar drugog panela. Jpanel kao platno za crtanje grafike.
`java.swing.JApplet`	Osnovna klasa za kreiranje Java apleta upotrebom Swing componenti.
`java.swing.JDialog`	Pop-up prozor kao privremeni prozor za primanje dodatnih informacija od korisnika ili radi obaveštavanja korisnika.

Slika 1.2 Kontejner klase u Javi

POMOĆNE GUI KLASE

Pomoćne klase služe da opišu svojstva GUI komponenata

Pomoćne klase nisu podklase klase Component. One služe da opišu svojstva GUI komponenata, kao što je grafički kontekst, boje, fontovi i dimenzije (Tabela 2).

Pomoćne klase su deo paketa ***java.awt. Swing*** klase ne zamenjuju sve***AWT*** klase, već AWT komponentne klase, tj. samo Button, TextField, TextArea i sl.

Tabela 2: Pomoćne klase

Pomoćne klase	Opis
`java.awt.Graphics`	Apstraktna klasa koja obezbeđuje metode za srtanje stringova, linija i jednostavnih objekata.
`java.awt.Color`	Podešava boje GUI komponenata.
`java.awt.Font`	Specificira fontove za tekstove i crteža u GUI komponentama.
`java.awt.FontMetrics`	Apstraktna klasa koja daje informaciju o svojstvima fontova.
`java.awt.Dimension`	Sadrži dimenzije širine i visine komponente (celi brojevi) GUO objekta.
`java.awt.LayoutManager`	Određuje raspored komponenata u nekom kontejneru.

Slika 1.3 Pomoćne klase u Javi

PRIMER 2 - PRIKAZ STRINGA U GRAFIČKOM KORISNIČKOM INTERFEJSU

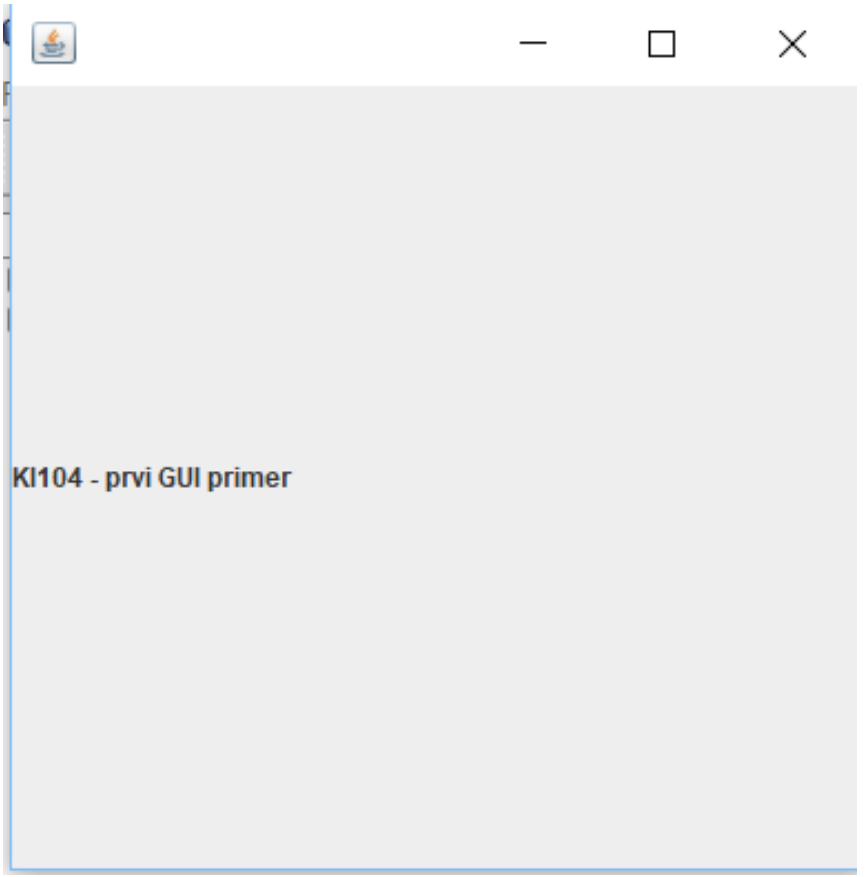
Prvi kompletan GUI program

Za prvu ilustraciju rada sa grafičkim korisničkim interfejsom u Javi biće kreiran i demonstriran jednostavan program. Program koristi jedan ***JFrame*** okvir na način da će glavna klasa da bude naslednica klase ***JFrame***. Nakon toga je kreiran je dan objekat klase ***JLabel*** (nalepnica) koji ima zadatak da na kreiranom okviru prikaže željeni tekst.

Navedene funkcionalnosti su realizovane sledećim programskim kodom:

```
package helloworldframe;

/**
 *
 * @author Vladimir Milicevic
 */
import javax.swing.JFrame;
import javax.swing.JLabel;


//Klasa nasleđuje JFrame
public class KI104Frame extends JFrame {

    public static void main(String args[]) {
       //poziv konstruktora
        new KI104Frame();
    }

   // konstruktor
    KI104Frame() {
        //kreiranje labele sa tekstom
        JLabel jlb = new JLabel("KI104 - prvi GUI primer");
       // dodavanje labele na okvir
        add(jlb);
       //dimenzije okvira: x i y
        this.setSize(400, 400);
        // podešavanje vidljivosti komponenata
        setVisible(true);
    }
}
```

Pokretanjem kreiranog primera dobija se izlaz kao na sledećoj slici.

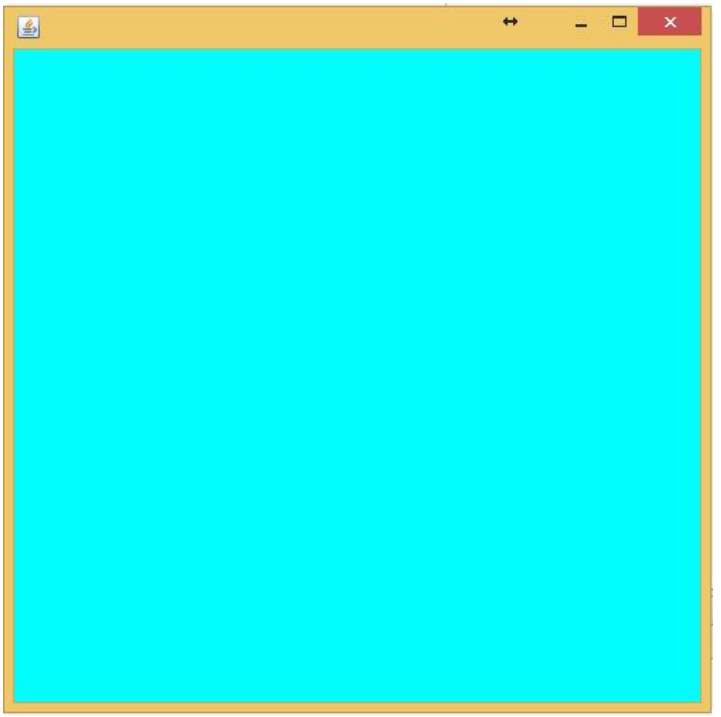


Slika 1.4 "HelloWorld" primer u grafičkom korisničkom interfejsu

PRIMER 3

Kreiranje proste forme u Java programskom jeziku.

Kreirati program na osnovu slike. Zadatak treba da bude deo projekta KI105-V11 i paketa zadatak1. Pokretačka klasa nosi naziv Zadatak1.



Slika 1.5 JPanel okvir

Objašnjenje:

Da bi se prikazala grafička komponenta, klasa Main u ovom slučaju nasleđuje klasu JFrame paketa java.swing koja predstavlja prozor u Javi. Za postavljanje pozadine samog prozora koristimo metodu getContentPane koja nam omogućava da dobijemo sadržaj samog prozora. Nakon toga metodom setBackground postavljamo novu boju pozadine, u ovom slučaju koristeći definisanu cyan boju klase Color, paketa java.awt.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package zadatak1;

import java.awt.Color;
import javax.swing.JFrame;

/**
 *
 * @author razvoj
 */
public class Zadatak1 extends JFrame {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        new Zadatak1();
```

```
    }

    public Zadatak1() {
        /**
         * Setujemo velicinu x 500 y 500
         */
        setSize(500, 500);
        /**
         * Centriramo formu u odnosu na ekran
         *
         */
        setLocationRelativeTo(null);
        /**
         * Setujemo boju na svetlo plavu
         */
        getContentPane().setBackground(Color.cyan);
        /**
         * zatvaranje aplikacije pritiskom na x dugme
         */
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        /**
         * Postavljamo formu da je vidljiva
         */
        setVisible(true);
    }

}
```

ZADATAK 1

Dopunite Primer 1

- Otvorite u razvojnom okruženju *NetBeans IDE* projekat koji odgovara primeru "Primer1" s prethodne sekcije;
- Kreirajte dve nove labele koje ćete dodati na okvir;
- Labele bi trebalo da prikazuju vaše podatke na ekranu;
- Dodajte nove labele na okvir;
- Pokrenite i testirajte modifikovani program.

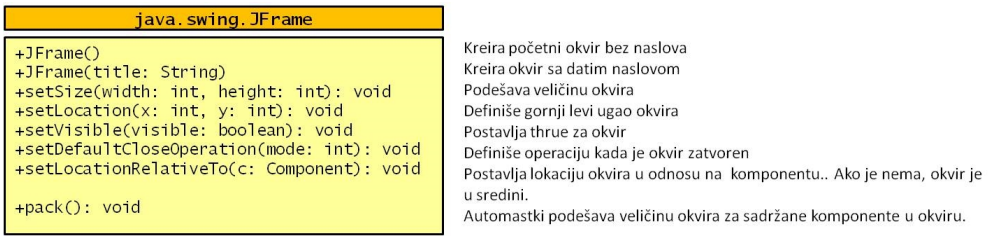
Poglavlje 2

Okviri (JFrame)

KLASA JFRAME

Okvir (frame) je prozor koji sadrži druge GUI komponente.

Da bi se kreirao korisnički interfejs, neophodno je da se kreira okvir ili aplet koji bi sadržao sve komponente interfejsa. Za kreiranje okvira koristi se klasa ***JFrame*** (slika 1)



Slika 2.1 Klasa JFrame

PRIMER: KORIŠĆENJE KLASE JFRAME

Okvir se ne prikazuje na monitoru dok se ne pozove metod frame.setVisible(true)

Okvir predstavlja objekat klase JFrame. Pozivanjem metoda *frame.setSize(400, 300)* određuje se širina okvira od 400 piksela, a visina od 300 piksela.

Pozivanjem metoda *setLocationRelativeTo(null)* centrira se okvir u odnosu na ekran.

Pozivanjem metoda *setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)* definiše da kada se okvir zatvori i aplikacija prekida sa radom. Ako se ovaj iskaz ne koristi, program ne staje kada se okvir zatvori. Da bi se program zaustavio, onda je potrebno pritisnuti Ctrl+C ili zaustaviti proces korišćenjem UNIX instrukcija. Ako izvršavate program u okviru Eclipse-a ili NetBeans-a trebalo bi pritisnuti dirku Terminate na konzoli da bi se zaustavio program.

Okvir se ne prikazuje na monitoru dok se ne pozove metod *frame.setVisible(true).*

Ako se*frame.setSize(400, 300)* i*setLocationRelativeTo(null)* metode ne pozovu, automatski se primenjuju metodi koji se nasleđuju iz klase JFrame.

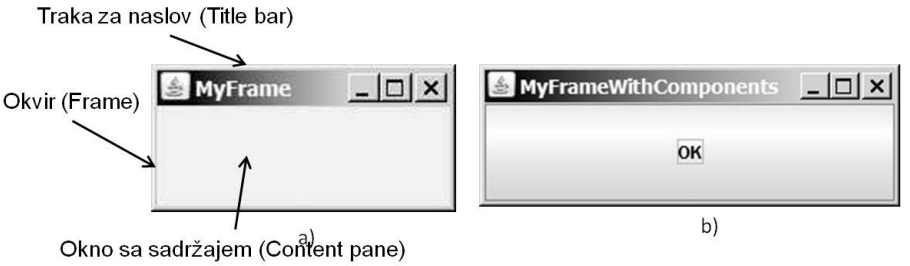
```
import javax.swing.JFrame;

public class MyFrame {
```

```
    public static void main(String[] args) {
        JFrame frame = new JFrame("MyFrame"); // Kreiranje okvira
        frame.setSize(400, 300); // Postavljanje veličine okvira
        frame.setLocationRelativeTo(null); // Cenar okvira
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true); // Prikaz okvira
    }
}
```

PRIMER 4 - KLASA MYFRAMEWITHCOMPONENTS

Svaki okvir (objekat klase JFrame) sadrži bar jedno okno za sadržaj (content pane) koji je objekat klase java.awt.Container.

Izvršenjem programa **MyFrame**, dobija se prozor na slici 2a, koji se pojavljuje na ekranu..



Slika 2.2 okvir sa naslovom komponente

```
import javax.swing.*;

public class MyFrameWithComponents {

    public static void main(String[] args) {
        JFrame frame = new JFrame("MyFrameWithComponents");

        // Add a button to the frame
        JButton jbtnOK = new JButton("OK");
        frame.add(jbtnOK);

        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null); // Centar okvira
        frame.setVisible(true);
    }
}
```

Svaki okvir (objekt klase ***JFrame***) sadrži bar jedno okno za sadržaj (engl., content pane) koji je primerak klase ***java.awt.Container***. GUI komponente, kao što su dugmad, na primer, postavljaju se u okno za sadržaj u okviru. Komponenta (na primer, dugme ***jbtnOK)*** se smešta u okvir pozivom metoda (tačnije rečeno, u okno za sadržaj okvira):

```
frame.add(jbtnOK);
```

Ovo svojstvo se naziva delegiranje okna za sadržaj. Na listingu klase **MyFrameWIthComponents** objekat klase ***JButton*** je kreiran sa new ***JButton(„OK“),*** i ovaj objekat je dodat oknu za sadržaj okvira (linija 9).

Metod **add(Component comp)** je definisan u klasi ***Container*** i on dodaje primerak (objekat) klase ***Component*** u kontejner. Kako je ***JButton*** podklasa klase ***Component,*** primerak klase ***JButton*** je takođe i primerak klase ***Content***. Da bi se komponenta uklonila iz kontejnera, upotrebite metod ***remove***. Sledeći iskaz uklanja dugme iz kontejnera:

```
Container.remove(jbtnOK);
```

Kada izvršavate program MyFrameComponents, prozor će se prikazati kao na slici 6b. Dugme je uvek u sredini okvira, bez obzira na njegove dimenzije, ako se drugačije ne odredi položaj digmeta.

ZADATAK 2

Dopunite primer 4

- Otvorite u razvojnom okruženju NetBeans IDE projekat koji odgovara primeru "Primer4" s prethodne sekcije;
- Kreirajte labelu koja prikazuje tekst "Kliknite na OK ili Cancel"
- Dodajte novu labelu na okvir;
- Kreirajte novo dugme koje prikazuje tekst "Cancel"
- Dodajte dugme na okvir;
- Pokrenite i testirajte modifikovani program.
- Pogledajte sledeću sekciju i pokušajte da lepo složiti komponente nekim menadžerom rasporeda.

UVOD U SWING GUI (VIDEO)

Video daje početne informacije u kreiranj prostora na displeju ya GUI.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

Poglavlje 3

Menadžeri rasporeda

MENADŽERI RASPOREDA I FLOWLAYOUT MENADŽER

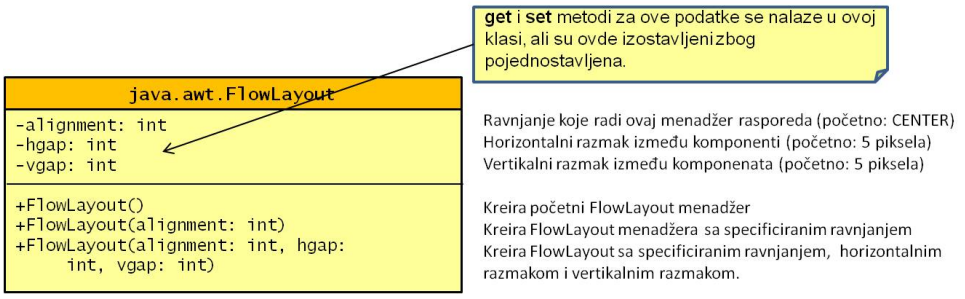
Svaki kontejner sadrži menadžera rasporeda, koji je predstavlja objekat koji je odgovoran za raspored GUI komponenata u kontejneru.

Menadžeri rasporeda komponenti obezbeđuju automatsko raspoređivanje komponenti korisničkog interfejsa na svim sistemima. Java GUI komponente se postavljaju u kontejnere, po rasporedu koji određuje menadžer rasporeda. Pozivom metoda **setLayout(aLayoutManager)** se postavljaju u kontejnere, kao na primer:

```
LayoutManager layoutManager = new XLayout();
container.setLayout(layoutManager);
```

Postoji tri osnovna menadžera rasporeda: ***FlowLayout, GridLayout, i BorderLayout.***

FlowLayout je najjednostavniji menadžer sadržaja. On ređa komponente u kontejneru s leva udesno, po redosledu njihovog dodavanja. Kada se popuni jedan red, stvara se novi red. Ravnjanje komponenata se oderditi upotrebom tri konstante: ***FlowLayout.RIGHT, FlowLayout.CENTER, ili FlowLayout.LEFT***. Na slici 1 prikazan je dijagrama klase FlowLayout.



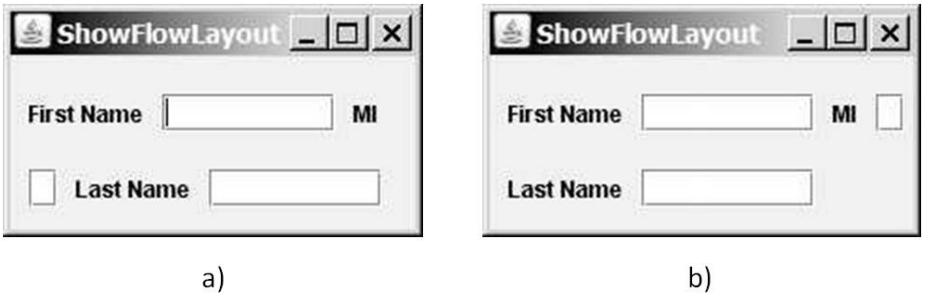
Slika 3.1 Klasa FowLayout

PRIMER 5 - FLOWLAYOUT MENADŽER RASPOREDA

FlowLayout menadžer raspoređuje komponente u kontejner po redovima

Dole je prikazan je listing programa koji pokazuje korišćenje menadžera rasporeda tip ***FlowLayout***. Program dodaje tri oznake (engl. labels) i tri polja za tekst (engl., text fields) prikazani na slici 2 .

```
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JFrame;
import java.awt.FlowLayout;

public class ShowFlowLayout extends JFrame {

    public ShowFlowLayout() {
        // Postavljanje  FlowLayout, levo ravnjanje sa horizontalnim razmakom
        // 10 i vertikalnim razmakom 20 između komponenti
        setLayout(new FlowLayout(FlowLayout.LEFT, 10, 20));

        // Dodavanje natpisa i polja za unos teksta u okvir
        add(new JLabel("First Name"));
        add(new JTextField(8));
        add(new JLabel("MI"));
        add(new JTextField(1));
        add(new JLabel("Last Name"));
        add(new JTextField(8));
    }

    /**
     * Main metod
     */
    public static void main(String[] args) {
        ShowFlowLayout frame = new ShowFlowLayout();
        frame.setTitle("ShowFlowLayout");
        frame.setSize(200, 200);
        frame.setLocationRelativeTo(null); // Center the frame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```



Slika 3.2 Izgled dobijenog okvira (a)

U ovom primeru, kreira se klasa **ShowFlowLayout** koja proširuje ***JFrame*** klasu (linija 6). Metod main kreira primerak klase ShowFlowLayout (linija 23). Konstruktor klase **ShowFlowLayout** konstruiše i raspoređuje komponente u okviru. Menadžer ***FlowLayout***

postavlja komponente u okviru. Ako se promeni veličina okvira, komponente se automatski raspoređuju u skladu sa novom veličinom okvira. Na slici 2a prvi red ima tri komponente, a na slici 2b, prvi red ima četiri komponente, jer je širina okvira povećana. Kreiran je anonimni objekat klase ***FlowLayout(FlowLayout.RIGHT, 0, 0)***), i izrazu na liniji 10, koji je ekvivalentan izrazu:

```
FlowLayout layout = new FlowLayout(FlowLayout.LEFT, 10, 20);
setLayout(layout);
```

Da li se u jedan okvir jedno dugme može dodati više puta? Ne, jer se jedno dugme (kao i sve GUI komponente) može dodati samo u jedan kontejner i to samo jedanput.

PRIMER 6

Cilj ovog zadatka je prikaz rada FlowLayout-a

Napraviti program koji će imati 6 dugmeta postavljenih kao na slici. Svako dugme treba da ima događaj da kada se pritisne ispisuje broj tog dugmeta. Zadatak treba da bude deo projekta KI105-V11 i paketa zadatak2. Pokretačka klasa nosi naziv Zadatak2.

Slika 3.3 Zadatak 2

Objašnjenje:

Kako je svaki klik zapravo ista akcija sa različitim parametrom koji se prikazuje, za obradu kreiramo klasu MyListener. Ova klasa implementira ActionListener i kao objekat će biti dodata svakom dugmetu sa slike.

Da bi se prikazao broj dugmeta na koji je kliknuto, u okviru bloka metode actionPerformed, pozivamo metodu getActionCommand() objekta ActionEvent.

Rešenje (Klasa Listener)

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package zadatak2;
```

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

/**
 *
 * @author razvoj
 */
public class MyListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        System.out.println(e.getActionCommand() + " was clicked");
    }

}
```

Rešenje (Klasa Main):

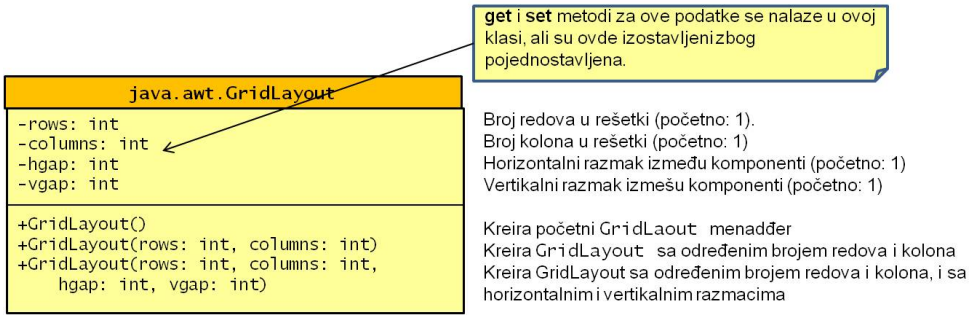
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package zadatak2;

import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

/**
 *
 * @author razvoj
 */
public class Zadatak2 extends JFrame {

    private JButton btn1, btn2, btn3, btn4, btn5, btn6;

    public static void main(String[] args) {
        new Zadatak2();

    }

    public Zadatak2() {
        setSize(450, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Main");
        setLocationRelativeTo(null); // Center the frame
        setVisible(true);
        setupGUI();
    }
```

```
    private void setupGUI() {

        JPanel p1 = new JPanel();
        p1.add(btn1 = new JButton("Button 1"));
        p1.add(btn2 = new JButton("Button 2"));
        p1.add(btn3 = new JButton("Button 3"));

        JPanel p2 = new JPanel();
        p2.add(btn4 = new JButton("Button 4"));
        p2.add(btn5 = new JButton("Button 5"));
        p2.add(btn6 = new JButton("Button 6"));
        setLayout(new FlowLayout());
        add(p1);
        add(p2);

        btn1.addActionListener(new MyListener());
        btn2.addActionListener(new MyListener());
        btn3.addActionListener(new MyListener());
        btn4.addActionListener(new MyListener());
        btn5.addActionListener(new MyListener());
        btn6.addActionListener(new MyListener());
    }

}
```

GRIDLAYOUT MENADŽER RASPOREDA

Menadžer GridLayout postavlja komponente po ćelijama jednake veličine na rešetki

Menadžer rasporeda ***GridLayout*** raspoređuje komponente po rešetki (matrici). Komponente se ređaju od levo ka desno, u prvom redu, a onda tako u drugom redu, itd. Dijagram klase za ***GridLayout*** je prikazan na slici 3 .



Slika 3.4 Menadžer GridLayout postavlja komponente po ćelijama jednake veličine na rešetki

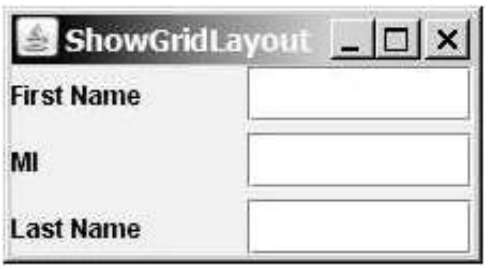
Pri slaganju komponenti u redove i kolone, primenjuju se sledeća pravila:

1. Broj redova ili kolona može biti nula, ali ne oba broja. Ako je jedan broj nula, a drugi različit od nule, onda je dimenzija koja je različita od nule fiksirana (ne menja se), a dimenzija koja je u početku nula se menja dinamički od strane menadžera rasporeda.
2. Ako su i broj redova i broj kolona različiti od nule, broj redova je dominirajući parametar. To znači da je njihov broj fiksan, a da menadžer rasporeda dinamički određuje potreban broj kolona.

PRIMER 7 - PRIMENA GRIDLAYOUT MENADŽERA RASPOREDA

Menadžer GridLayout deli kontejner u rešetku, a komponente dodavanjem popunjavaju ćelije rešetke, red po red.

Na slici 4 prikazan je izgleda okovira koji je dobijen programom čiji je listing ovde prikazan. On pokazuje primenu rasporeda GUI komponenata po rešetki. Program je sličan programu koji je dat za slučaj FlowLayout menadžera rasporeda, samo što dodaje tri oznake i tri tekstualna polja u okviru***GridLayout*** umesto u okviru ***FlowLayout.***



Slika 3.5 Menadžer GridLayout deli kontejner u rešetku, a komponente dodavanjem popunjavaju ćelije rešetke, red po red

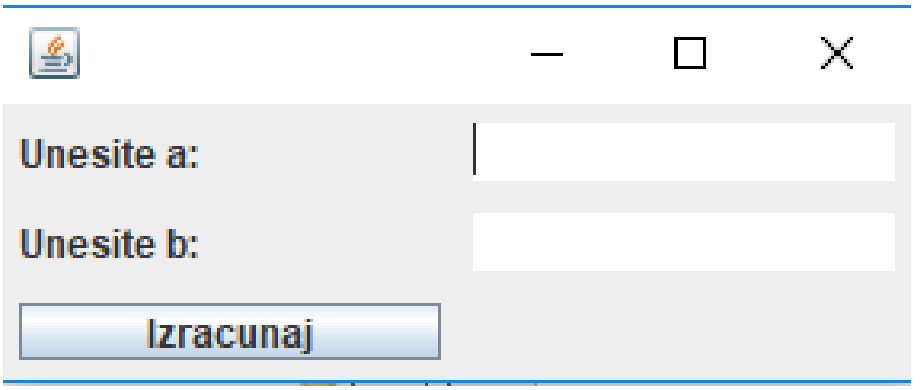
```
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JFrame;
import java.awt.GridLayout;

public class ShowGridLayout extends JFrame {

    public ShowGridLayout() {
        // Postavlja  GridLayout, 3 reda, 2 kolone, i razmak 5 između
        // komponenti i po horizontali i po vertikali
        setLayout(new GridLayout(3, 2, 5, 5));

        // Dodavanje natpisa i polja za tekst  u okvir
        add(new JLabel("First Name"));
        add(new JTextField(8));
        add(new JLabel("MI"));
        add(new JTextField(1));
        add(new JLabel("Last Name"));
        add(new JTextField(8));
    }
```

```
    /**
     * Main metod
     */
    public static void main(String[] args) {
        ShowGridLayout frame = new ShowGridLayout();
        frame.setTitle("ShowGridLayout");
        frame.setSize(200, 125);
        frame.setLocationRelativeTo(null); // Center the frame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);

    }
}
```

PRIMER 8

Cilj ovog zadatka je pravljenje prostog programa za sabiranje dva broja sa grafičkim intefejsom u programskom jeziku Java.

Kreirati program na osnovu slike. Klikom na dugme *Izračunaj,* program treba da sabere dva broja. Zadatak treba da bude deo projekta KI105-V11 i paketa zadatak3. Pokretačka klasa nosi naziv Zadatak3.



Slika 3.6 Zadatak 3

Objašnjenje:

GridLayout nam omogućava da komponente ređamo po kolonama i redovima. Grid Layout kroz konstruktor prima broj redova, broj kolona, vertikalni i horizontalni razmak između elemenata. Pri dodavanju komponenata na panel, grid menadžer rasporeda ih sam ređa po principu (0,0) (0,1) (1,0) (1,1) ukoliko npr. imamo dve kolone i dva reda.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package zadatak3;

import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

/**
 *
 * @author razvoj
 */
public class Zadatak3 extends JFrame {

    JPanel panel = new JPanel();
    JButton btn = new JButton("Izracunaj");
    JTextField unos1 = new JTextField();
    JTextField unos2 = new JTextField();
    JLabel label1 = new JLabel("Unesite a: ");
    JLabel label2 = new JLabel("Unesite b: ");

    public static void main(String[] args) {
        new Zadatak3();
    }

    public Zadatak3() {
        setSize(300, 125);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
        /**
         * Koristimo grid layout kao manager rasporeda sa 3 reda i 2 kolone, pri
         * čemu je horizontalni i vertikalni razmak 10
         */
        panel.setLayout(new GridLayout(3, 2, 10, 10));
        setContentPane(panel);
        setupButtons();
    }

    public void setupButtons() {
        /**
         * Oko panela kreiramo prazan okvir kako bi se komponente odvojile od
         * ivica, u ovom slucaju po 5 od svake
         */
        panel.setBorder(new EmptyBorder(new Insets(5, 5, 5, 5)));
        panel.add(label1);
        panel.add(unos1);
```

```
        panel.add(label2);
        panel.add(unos2);
        panel.add(btn);
        panel.revalidate();
        btn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(rootPane, "Rezultat je : " +
(Double.parseDouble(unos1.getText()) + Double.parseDouble(unos2.getText())));

            }
        });
        panel.add(btn);
    }

    public void clickMe() {
    }

}
```

PRIMER 9

Cilj ovog zadatka je povezivanje klase sa grafikom sa klasom koja vrši logiku.

Kreirati program na osnovu slike. Klikom na dugme Izračunaj, program treba da na osnovu unetog x računa x^3+4x. Zadatak treba da bude deo projekta KI105-V11 i paketa zadatak4. Pokretačka klasa nosi naziv Zadatak4.

Slika 3.7 Zadatak 4

Objašnjenje:

U okviru klika na dugme *Izračunaj* pozivamo metodu kojoj prosleđujemo parametar pročitan sa GUI-ja, koja ima zadatak da preračuna i vrati rezultat na osnovu zadate formule.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
```

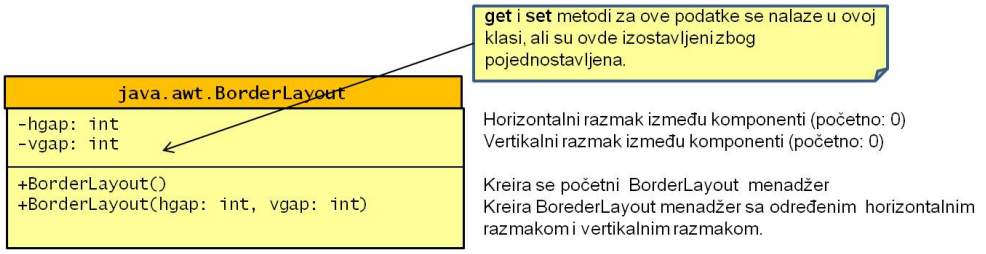
```
 * and open the template in the editor.
 */
package zadatak4;

import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

/**
 *
 * @author razvoj
 */
public class Zadatak4 extends JFrame {

    JPanel panel = new JPanel();
    JButton btn = new JButton("Izracunaj");
    JTextField unos1 = new JTextField();
    JLabel label1 = new JLabel("Unesite x: ");

    public static void main(String[] args) {
        new Zadatak4();
    }

    public Zadatak4() {
        setSize(300, 125);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
        panel.setLayout(new GridLayout(2, 2, 10, 10));
        setContentPane(panel);
        setupGUI();
    }

    public void setupGUI() {
        /**
         * Oko panela kreiramo prazan okvir kako bi se komponente odvojile od
         * ivica, u ovom slucaju po 5 od svake
         */
        panel.setBorder(new EmptyBorder(new Insets(5, 5, 5, 5)));
        panel.add(label1);
        panel.add(unos1);
        panel.add(btn);
        btn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
```

```
                JOptionPane.showMessageDialog(rootPane, "Rezultat je : "
                        + izracunajRezultat(Double.parseDouble(unos1.getText())));
            }

        });
        panel.add(btn);
    }

    private double izracunajRezultat(double x) {
        return Math.pow(x, 3) + 4 * x;
    }
}
```

BORDERLAYOUT MENADŽER RASPOREDA

Menadžer GridLayout deli kontejner u rešetku, a komponente dodavanjem popunjavaju ćelije rešetke, red po red

Menadžer rasporeda ***BorderLayout*** deli kontejner na pet delova: istočni i – East, južni – South,zapadni – ***West***, severni – ***North*** i centar – ***Center***. Komponente se dodaju u kontejner primenom metoda add((Component index), gde je index jedna od mogućih konstanti: ***BorderLayout.EAST***, ***BorderLayout.SOUTH***, ***BorderLayout.WEST***, ***BorderLayout.NORTH***, ili ***BorderLayout.CENTER***. Slika 5 prikazuje dijagram klase ***BorderLayout.***



SLika-5: Menadžer rasporeda BorderLayout raspoređuje komponente u pet delova

Komponente se raspoređuju u skladu sa njihovim poželjnim veličinama i naznačenom delu kontejnera (slika 6). Komponente na severnom i južnom delu se šire po horizontali, a komponenta na istoku i zapadu se šire vertikalno. Komponenta u centru se širi i horizontalno i vertikalno. Širenje ili skupljanje ovih komponenti se automatski vrše pri promeni veličine okvira, odn. njegovog kontejnara.

SLika-6: Menadžer BorderLayout deli kontejner u pet područja i svako sadrži po jednu komponentu
Menadžer GridLayout deli kontejner u rešetku, a komponente dodavanjem popunjavaju ćelije rešetke, red po red

PRIMER 10 - PRIMENA BORDERLAYOUT MENADŽERA RASPOREDA

Komponente se raspoređuju u skladu sa njihovim poželjnim veličinama i naznačenom delu kontejnera

Listing programa daje program koji primenjuje dejstvo menadžera ***BorderLayout.*** Program raspoređuje pet dugmadi (slika 4) označeni sa East, South, West, North, iCenteru okviru sa ***BorderLayout*** menadžerom.

GUI komponente u obliku dugmadi (***JButton***) sa dodaju u okvir u linijama 11-15. Ovde je metod add() drugačiji od metoda **add()** kod ***FlowLayout*** i ***GridLayout*** menadžera. U slučaju primene ***BorderLayout*** menadžera vi određujete gde želite da postavite komponente (u jednu od pet delova okvira).

Nepotrebno je postavljati komponente tako da zauzmu sve delove okvira. Ako uklonite East dugme, u gornjem primeru, dugme Center će se proširiti tako da zauzme mesto gde je bilo dugme East.

Ako ne navodite mesto gde želite da postavite komponentu, menadžer ***BorderLayout*** podrazumeva da želite da je stavite u centar. Ako na taj način unosite više komponenti, samo će se poslednja postaviti u centar. Ostale se ne prikazuju na ekranu.

```
import javax.swing.JButton;
import javax.swing.JFrame;
import java.awt.BorderLayout;

public class ShowBorderLayout extends JFrame {

    public ShowBorderLayout() {
        // BorderLayout sa horizontalnim razmakom 5, a vertikalnim razmakom 10
        setLayout(new BorderLayout(5, 10));

        // Dodavanje dugmadi u okvir
        add(new JButton("East"), BorderLayout.EAST);
        add(new JButton("South"), BorderLayout.SOUTH);
        add(new JButton("West"), BorderLayout.WEST);
        add(new JButton("North"), BorderLayout.NORTH);
        add(new JButton("Center"), BorderLayout.CENTER);
    }

    /**
     * Main metod
     */
    public static void main(String[] args) {
        ShowBorderLayout frame = new ShowBorderLayout();
        frame.setTitle("ShowBorderLayout");
```

```
        frame.setSize(300, 200);
        frame.setLocationRelativeTo(null); // Center the frame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

PRIMER 11

Cilj ovog zadatka je prikaz korišećenja BorderLayout-a.

Kreirati program na osnovu slike, a koji ima 5 dugmeta. Zadatak treba da bude deo projekta KI105-V11 i paketa zadatak5. Pokretačka klasa nosi naziv Zadatak5.

Slika 3.8 Zadatak 5

Komponente na slici apsolutno odgovaraju rasporedu koji primenjuje BorderLayout i koji nam omogućava da koponente smeštamo u pozicije:

Gore (BorderLayout.PAGE_START)

Dole (BorderLayout.LINE_END)

Levo (BorderLayout.LINE_START)

Desno (BorderLayout.PAGE_END)

Centar (BorderLayout.CENTER)

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package zadatak5;

import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

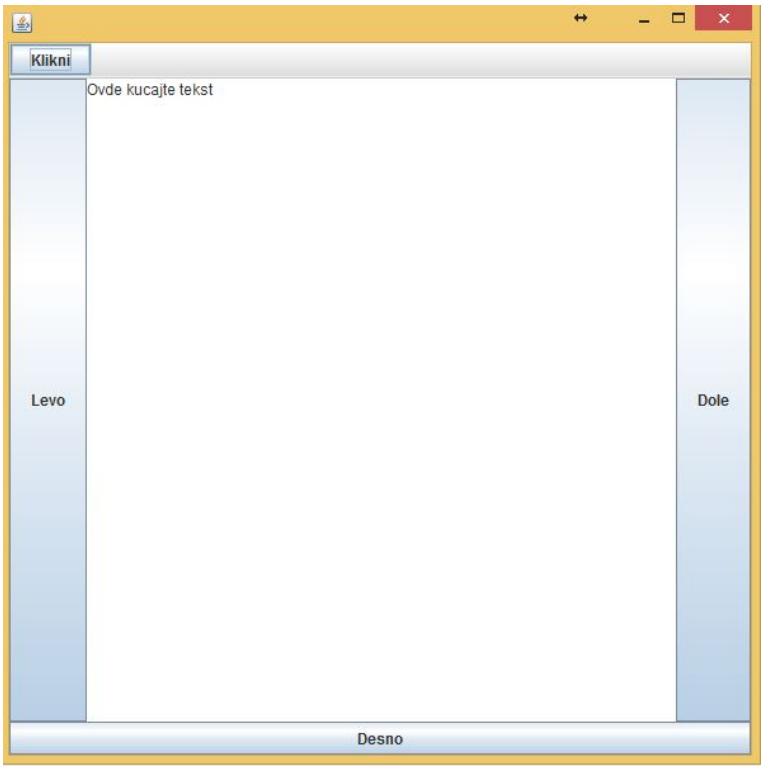
```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

/**
 *
 * @author razvoj
 */
public class Zadatak5 extends JFrame {

    JPanel panel = new JPanel();

    public static void main(String[] args) {
        new Zadatak5();
    }

    public Zadatak5() {
        setSize(300, 125);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
        panel.setLayout(new BorderLayout());
        setContentPane(panel);
        setupGUI();
    }

    public void setupGUI() {
        panel.add(new JButton("Header"), BorderLayout.PAGE_START);

        /**
         * Jedino centralno dugme ima odgovarajuću akciju na klik. U ovom
         * slučaju, to je poziv metode clickMe, koja štampa poruku Hello
         */
        JButton center = new JButton("Cetnralno dugme");
        center.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                clickMe();
            }
        });
        panel.add(center, BorderLayout.CENTER);

        panel.add(new JButton("Levo"), BorderLayout.LINE_START);
        panel.add(new JButton("Desno"), BorderLayout.PAGE_END);
        panel.add(new JButton("Dole"), BorderLayout.LINE_END);

    }

    public void clickMe() {
        JOptionPane.showMessageDialog(rootPane, "Hello");
    }
```

```
}
```

PRIMER 12

Cilj ovog zadatka je da prilagodimo grafički interfejs prethodnog zadatka tako da koristimo još i JTextArea i JMenu

Prilagoditi grafički interfejs prošlog zadatka tako da gore imamo meni sa jednim dugmetom i u centru imamo JTextArea kao na slici. Zadatak treba da bude deo projekta KI105-V11 i paketa zadatak6. Pokretačka klasa nosi naziv Zadatak6.



Slika 3.9 Zadatak 6

Objašnjenje:

Potrebno je samo na odgovarajućim mestima u prethodnom zadatku promeniti komponentu JButton u zadate na slici.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package zadatak6;

import java.awt.BorderLayout;
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
import javax.swing.JMenuBar;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import zadatak5.Zadatak5;

/**
 *
 * @author razvoj
 */
public class Zadatak6 extends JFrame {

    JPanel panel = new JPanel();

    public static void main(String[] args) {
        new Zadatak6();
    }

    public Zadatak6() {
        setSize(500, 500);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
        panel.setLayout(new BorderLayout());
        setContentPane(panel);
        setupGUI();
    }

    public void setupGUI() {
        JMenuBar menu = new JMenuBar();
        JButton meniButton = new JButton("Klikni");
        menu.add(meniButton);
        panel.add(menu, BorderLayout.PAGE_START);

        panel.add(new JTextArea("Ovde kucajte tekst"), BorderLayout.CENTER);

        panel.add(new JButton("Levo"), BorderLayout.LINE_START);
        panel.add(new JButton("Desno"), BorderLayout.PAGE_END);
        panel.add(new JButton("Dole"), BorderLayout.LINE_END);
        revalidate();
    }

}
```

SVOJSTVA MENADŽERA RASPOREDA

Ako želite da ta svojstva dinamički menjate (u samom programu, posle kreiranja), morate da menadžeru rasporeda dodelite ime i da je tretirate kao promenljivu

Menadžeri rasporeda imaju svojstva koja se mogu dinamički menjati:

1. ***FlowLayout*** – ima ***alignment, hgap i vgap*** svojstva. Sa metodima ***setAlignment, setHgap i setVgap*** možete ih menjati.
2. ***GridLayout*** – ima ***rows, columns, hgap i vgap*** svojstva. Za njihovu promenu koriste se metodi: ***setRows, setColumns, setHgap, i setVgap***
3. ***BorderLayout*** – ima ***hgap i vgap*** svojstva. Ona se menjaju metodima ***setHgap i setVgap***.

U prikazanim primerima koristili smo anonimne menadžera rasporeda jer se njihova svojstva ne menjaju od momenta njihovog kreiranja. Ako želite da ta svojstva dinamički menjate (u samom programu, posle kreiranja), morate da menadžeru rasporeda dodelite ime i da je tretirate kao promenljivu. Ovo se vidi u sledećem primeru:

```
// Kreiranje Layout menadžera
FlowLayout flowLayout = new FlowLayout();
// Postavljanje svojstava Layout menadžer
flowLayout.setAlignment(FlowLayout.RIGHT);
flowLayout.setHgap(10);
flowLayout.setVgap(20);
```

MENADŽERI RASPOREDA U JAVI SWING PAKETU (VIDEO)

Video prikazuje menadćere rasporeda u Swing paketu

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

ZADACI 3 - 8

Samostalno vežbanje rada sa menadžerima rasporeda.

Zadatak 3. Da li će program ShowFlowLayout raditi ako se u liniji 23 ShowFlowLayout zameni sa JFrame? Da li će program ShowGridLayout raditi ako se u liniji 23 ShowGridLayout zameni sa JFrame? Da li će program ShowBorderLayout raditi ako se u liniji 20 ShowBorderLayout zameni sa JFrame?

Zadatak 4. Zašto vam je potreban menadžer rasporeda? Koji je unapred definisani menadžer rasporeda za okvir? Kako dodajete komponente u okvir?

Zadatak 5. Opišite FlowLayout. Kako kreirate FlowLayout menadžer? Kako dodajete komponentu u kontejner kojim upravlja menadžer FlowLayout? Koja su svojstva kojim se regulišu horizontalni i vertikalni razmaci između komponenti u kontejneru? Da li možete da definišete ravnjanje?

Zadatak 6. Opišite GridLayout. Kako kreirate GridLayout menadžer? Kako dodajete komponentu u kontejner kojim upravlja menadžer GridLayout? Da li ima ograničenja broja

komponenti koja možete dodati u GridLayout kontejner? Koja su svojstva kojim se regulišu horizontalni i vertikalni razmaciji između komponenti u kontejneru?

Zadatak 7. Opišite BorderLayout. Kako kreirate BorderLayout menadžer? Kako dodajete komponentu u kontejner kojim upravlja menadžer BorderLayout? Koja su svojstva kojim se regulišu horizontalni i vertikalni razmaci između komponenti u kontejneru?

Zadatak 8. Sledeći program bi trebalo da prikaže jedno dugme u okviru, ali ono nije prikazano na ekranu. Šta je problem?

```
public class Test extends javax.swing.JFrame {

    public Test() {
        add(new javax.swing.JButton("OK"));
    }

    public static void main(String[] args) {
        javax.swing.JFrame frame = new javax.swing.JFrame();
        frame.setSize(100, 200);
        frame.setVisible(true);
    }
}
```

PROGRAMIRANJE SWING GUI (VIDEO)

Video pokazuje početna znanja neophpdna za kreiranje GUI (grafičkog korisničkog interfeksa).

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

Poglavlje 4

Upotreba panela kao podkontejnera

JPANEL KLASA

Kontejner se može postaviti unutar drugog kontejnera, te predstavlja podkontejner. Paneli se koriste kao podkontejneri radi grupisanja GUI komponenti da bi se ostvario željeni raspored.

Pretpostavimo da želite da postavite 10 dugmadi i tekstualnih polja u okvir vašeg GUI-a. Dugmad se postavljaju po rešetki, a tekstualna polja se postavljaju u jednom redu. Ovo je teško realizovati ako se koristi samo jedan kontejner. Mnogo lakše se ostvaruje ako se prozor, tj. okvir podeli na nekoliko panela. Paneli imaju ulogu podkontejnera koji sadrže grupe GUI komponenti. Možete dodati dugmad u jedan panel, a onda d apanel dodate u okvir. U drugi panel možete da stavite tekstualna polja, pa njega da dodate u okvir.

U Swing paketu za rad sa panelima se koristi klasa***JPanel***, koji se kreira sa **new JPanel().** Kada se koristi unapred definisan raspored, onda je to **FlowLayout** menadžer. Ako kreirate panel sa **new JPanel(LayoutManager)** onda kreirate panel sa tačno navedenim menadžerom rasporeda. Koristite metod **add(Component)** da bi dodali komponentu u panel. Na primer, sledeći kod kreira panel i dodaje jedno dugme u taj panel:

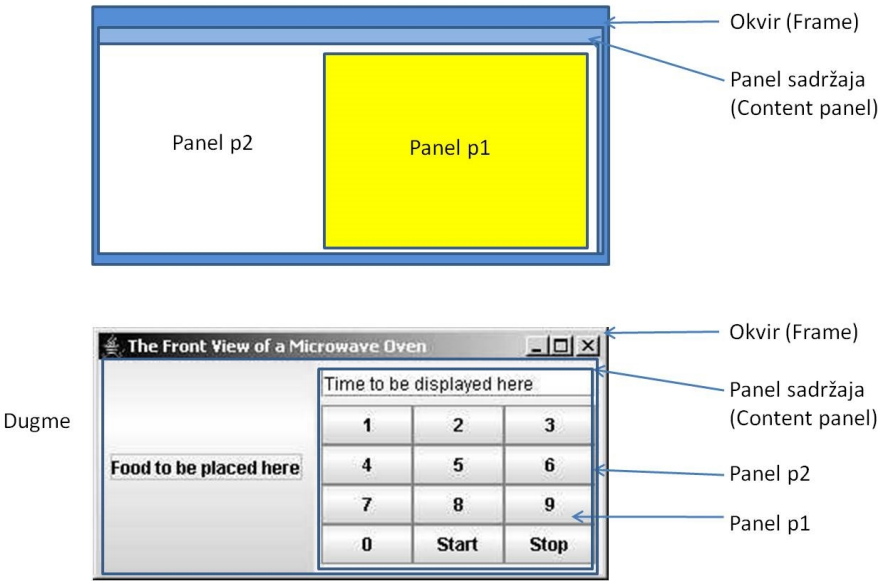
```
JPanel p = new JPanel();
p.add(new JButton("OK"));
```

Paneli se mogu ubaciti u jedan okvir ili u neki drugi panel. Za postavljanje panela f u okvir p trebalo bi da koristite sledeći iskaz:

Paneli se mogu ubaciti u jedan okvir ili u neki drugi panel. Za postavljanje panela f u okvir p trebalo bi da koristite sledeći iskaz:

```
add(p);
```

Na slici 1 prikazana je upotreba panela kao kontejnera kao dela grafičkog korisničkog interfejsa za slučaj mikrotalasne peći.



Slika 4.1 Primer dobijenog GUI primenom dva panela

PRIMER 13 - PRIMENA JPANEL KLASE

Klasa Container je superklasa svih klasa koje predstavljaju GUI komponente, kao na primer, komponenta JButton. Svaka GUI komponenta je zbog toga kontejner

Listing prikazuje program za kreiranje grafičkog korisničkog interfejsa sa dva panela, prikazanog na slici 1.

```
import java.awt.*;
import javax.swing.*;

public class TestPanels extends JFrame {

    public TestPanels() {
        // Kreiranje panela p1 za dugmad i postavljanje GridLayout menadžera
        JPanel p1 = new JPanel();
        p1.setLayout(new GridLayout(4, 3));

        // Dodavanje dugmadi na panel
        for (int i = 1; i <= 9; i++) {
            p1.add(new JButton("" + i));
        }

        p1.add(new JButton("" + 0));
        p1.add(new JButton("Start"));
        p1.add(new JButton("Stop"));

        // Kreiranje panela p2 za smeštaj tekstualnih polja  i panela p1
        JPanel p2 = new JPanel(new BorderLayout());
        p2.add(new JTextField("Time to be displayed here"),
```

```
                BorderLayout.NORTH);
        p2.add(p1, BorderLayout.CENTER);

        // Dodavanje sadržaja u okvir
        add(p2, BorderLayout.EAST);
        add(new JButton("Food to be placed here"),
                BorderLayout.CENTER);
    }

    /**
     * Main metod
     */
    public static void main(String[] args) {
        TestPanels frame = new TestPanels();
        frame.setTitle("The Front View of a Microwave Oven");
        frame.setSize(400, 250);
        frame.setLocationRelativeTo(null); //  Centar okvira
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Metod ***setLayout je*** definisan u klasi ***java.awt.Container***. Kako je ***JPanel*** podklasa klase ***Container***, možete koristiti setLayout da bi postavili željeni menadžer rasporeda (linija 8). Linije 7-8 mogu da se zamene sa iskazom:

```
JPanel p1 = new JPanel(new GridLayout(4, 3)).
```

Da bi se dobio željeni interfejs, program upotrebljava panel p1 da grupiše dugmad sa brojevima, Stop i Start dugmad, a panel p2 od ***BorderLayou***t da sadrži tekstualno polje u severnom delu panela p2, a panel p1 u centru panela p2. Dugme koje predstavlja hranu je postavljeno u centar okvira, a panel p2 je postavljen na istočni deo okvira. Iskaz (linije 21-22)

```
p2.add(new JTextField("Time to be displayed here"),
BorderLayout.NORTH);
```

Kreira primerak klase ***JTextField*** i dodaje ga u panel p2. GUI komponenta može se upotrebiti za unos tekstualnih informacija ili za prikazivanje vrednosti.

Klasa ***Containe***r je superklasa svih klasa koje predstavljaju GUI komponente, kao na primer, komponenta ***JButton.*** Svaka GUI komponenta je zbog toga kontejner. Teorijski gledano, zbog toga, možete koristiti metod ***setLayout*** i da rasporedite komponente i u jednom dugmetu, dodajući mu druge komponente, jer klasa ***JButton*** nasleđuje metode iz klase ***Container***. Međutim, iz praktičnih razloga ne bi trebalo da koristite dugmad kao kontejnere

ZADACI 9 - 17

Samostalno vežbanje upotrebe panela.

Zadatak 9. Kako kreirate panel sa specificiranim menadžerom rasporeda?

Zadatak 10. Koji je unapred definisan menadžer rasporeda za ***JPanel***? Kako dodajete neku komponentu u ***JPane***l?

Zadatak11. Da li možete upotrebiti metod ***setTitle*** u panelu? Koja je svrha upotrebe panela?

Zadatak 12. Kako je GUI komponenta klasa, kao što je ***JButton*****,** koja je podklasa klase ***Container,*** Da li možete dodati komponente u dugme?

Zadatak 13. UpotrebaFlowLayout menadžera: Napišite program koji zadovoljava sledeće zahteve (slika V1):

a) Kreira okvir i postavlja raspored pomoću ***FlowLayout*** menadžera.

b) Kreiranje dva panela i dodati ih u okvir.

c) Svaki panel sadrži po tri dugmadi. Paneli koriste ***FlowLayou***t menadžer.

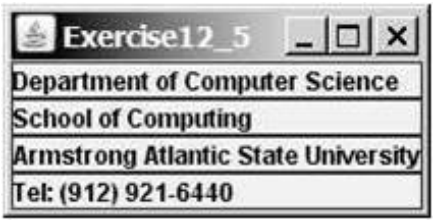
Slika 4.2 Zadati raspored komponenata

Zadatak 14. Upotreba BorderLayout menadžera: Napišite prethodni program za izradu istog korisničkog interfejsa, ali, umesto FlowLayout sa okvir, upotrebom BorderLayout. Postavite jedan panel na južni deo okvira a drugi – u centar.

Zadatak15. Upotreba GridLayout menadžera: Izmenite program iz zadatka 1 dodavanjem šest dugmadi u okvir. Upotrebite GridLayout u dva reda i u tri kolone za okvir.

Zadatak16 .Upotreba JPanel grupirana dugmadi: Izmenite program iz zadatka 1 da bi dobili isti interfejs, samo, umesto kreiranjem posebnih dugmadi i panela, definišite klasu koje proširuje klasu JPanel. Postavite tri dugmeta u vaš panel, i kreirajte dva panela koristeći klasu pnalea koji definiše korisnik.

Zadatak17. Prikaz natpisa: Napišite program koji prikazuje četiri linije teksta u četiri natpisa, kao što je prikazano na slici V2. Dodajte graničnu liniju oko svakog natpisa.



Slika 4.3 Prikaz natpisa

JPANEL (VIDEO)

Video objašnjava primenu JPanel

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

Poglavlje 5

Klase Color i Font

KLASA COLOR

Svaka GUI komponenta ima boju za pozadinu i za prednji prikaz. Voje su objekti kreirani iz klase Color.

Upotrebom klase***java.awt.Color*** možete podesiti boje neke GUI komponente. Svaka boja je mešavina tri osnovne boje: crvene, zelene i plave. Svaka od njih je predstavljena jednom int vrednošću koja opisuje njen intenzitet, u opsegu 0 (najtamnija nijansa) do 255 (najsvetlija nijansa).

Boja se kreira konstruktorom:

```
public Color(int r, int g, int b);
```

gde r, g i b predstavljaju crvenu, zelenu i plavu osnovnu boju, tj. komponentu određene boje, na primer:

```
Color color = new Color(128, 100, 100);
```

Možete koristiti metode ***setBackground(Color c)*** i ***setForeground(Color c)*** definisane u ***klasi java.awt.Component*** da bi odredili boju pozadine i prednjeg prikaza GUI komponenete. Evo jednog primera:

```
JButton jbtOK = new JButton("OK");
jbtOK.setBackground(color);
jbtOK.setForeground(new Color(100, 1, 1));
```

Drugi način da definišete boje je da upotrebite jednu od unapred definisanih 13 boja (BLACK, BLUE, CYAN, DARK_GRAY, GRAY, GREEN, LIGHT_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, and YELLOW) koje su definisane kao konstante u ***java.awt.Color***. Na primer, sledeći iskaz na ovaj način definiše crvenu boju dugmeta.

```
jbtOK.setForeground(Color.RED);
```

KLASA FONT

Svaka GUI komponenta ima svojstvo fonta. Fontovi su objekti kreirani pomoću klase Font.

Možete kreirati font upotrebom java.awt.Font klase i da sa metodom ***setFont ()*** postavite fontove komponenata. Konstruktor za klasu ***Font*** je:

```
public Font(String name, int style, int size);
```

Možete izabrati sledeće fontove: ***SansSerif, Serif, Monospaced, Dialog, i DialogInput,*** Stil fonta se bira pozivom:***Font.PLAIN (0), Font.BOLD (1), Font.ITALIC (2), i Font.BOLD + Font.ITALIC (3***), pri čemu se unosi i veličina fonta kao pozitivni ceo broj. Na primer, sledeći iskazi kreiraju dva fonta a jedan font se definiše samo za dugme:

```
Font font1 = new Font("SansSerif", Font.BOLD, 16);
Font font2 = new Font("Serif", Font.BOLD + Font.ITALIC, 12);
JButton jbtOK = new JButton("OK");
jbtOK.setFont(font1);
```

Ako vaš sistem ima i druge fontove, možete ih koristiti radi kreiranja Font objekta. Da bi našli koje fontove obezbeđuje vaš sistem treba da dobijete primerak klase: java.awt.GraphicsEnvironment upotrebom statičkog metoda: getLocalGraphicsEnvironment()

GraphicsEnvironment je apstraktna klasa koja opisuje grafičko okruženje određenog sistema. Možete koristite metod getAllFonts() da bi dobili sve raspoložive fontove sistema na kome radite. Koristite metod ***getAvailableFontFamilyNames()*** da bi dobili imena svi mogućih fontova. Na primer, sledeći iskazi štampaju imena svih raspoloživih fontova na sistemu:

```
GraphicsEnvironment e =
GraphicsEnvironment.getLocalGraphicsEnvironment();
String[] fontnames = e.getAvailableFontFamilyNames();
for (int i = 0; i < fontnames.length; i++)
System.out.println(fontnames[i]);
```

JCOLORCHOOSER (VIDEO)

Video objašnjava biranje i korišćenje boja.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

IZBOR I KORIŠĆENJE BOJA (VIDEO)

Primena klase JColorChooser

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

PRIMER 14 - PRIMENA KLASA FONT I COLOR

Primena klasa Font i Color

Kreirati program koji na labeli podešava font po izboru, a boju pozadine takođe podešava po vlastitom izboru.

Koristiti klase ***Font*** i ***Color*** za realizaciju navedenih funkcionalnosti.

Sledi listing klase *FontColorPrimer.java*.

```
package fontcolorprimer;

/**
 *
 * @author Vladimir Milicevic
 */

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;

import javax.swing.JFrame;
import javax.swing.JLabel;

public class FontColorPrimer {

    public static void main(String[] args) {
        //ulepšavanje prikaza okvira
        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new JFrame();
        //podešavanje naziva okvira
        frame.setTitle("JLabel Test");
        //podešavanje menadžera rasporeda
        frame.setLayout(new FlowLayout());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //nova labela
        JLabel label = new JLabel("Ime");
        //podešavanje fonta na labeli
        label.setFont(new Font("Courier New", Font.ITALIC, 12));
        //podešavanje boje pozadine
        label.setForeground(Color.GRAY);

        frame.add(label);
        //podešavanje veličine okvira u zavisnosti od komponenata, na
        //najmanju dozvoljenu veličinu
        frame.pack();
        frame.setVisible(true);
    }
}
```

Obratite pažnju na komentare u kodu kojima su detaljno objašnjene odgovarajuće linije koda.

Izvršavanju prikazanog listinga, odgovara sledeća slika.

Slika 5.1 Primena klasa Font i Color

ZADACI 10 - 14

Samostalno vežbanje klasa Color i Font

Zadatak 10. Kako kreirate boju? Šta je pogrešno u vezi kreiranja boje upotrebom **new Color(400, 200, 300)?** Koja je boja tamnija od ove dve boje: **new Color(10, 0, 0) ili new Color(200, 0, 0)?**

Zadatak11. Kako kreirate objekat ***Color*** sa slučajnom bojom?

Zadatak 12. Kako podešavate da objekt dugme ***JbtOK*** ima sa plavu pozadinu?

Zadatak 13. Kako kreirate ***Font*** objekat sa imenom fonta ***Courie***r, veličinom 20 i stilom bold?

Zadatak 14. Kako nalazite sve raspoložive fontove na vašem sistemu?

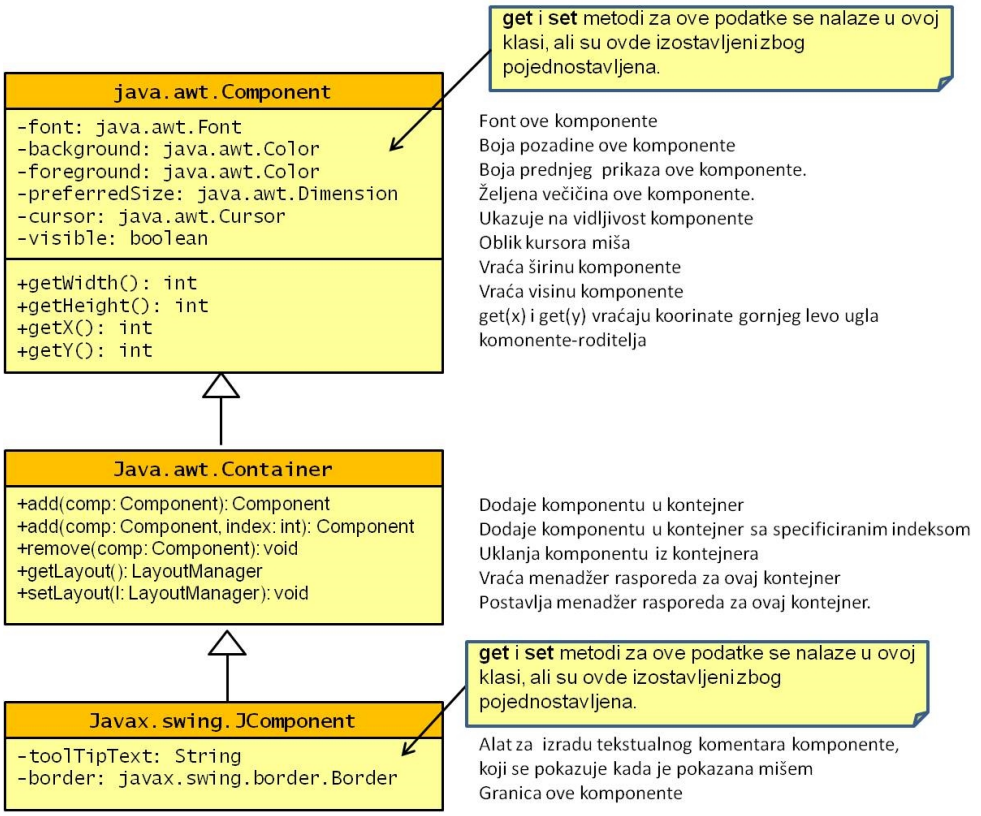
Poglavlje 6

Zajednička svojstva Swing GUI komponenata

KLASE COMPONENT, CONTAINER I JCOMPONENT

GUI komponente imaju neka zajednička svojstva. Oni su definisani u superklasama Component, Container i JComponent

Klasa ***Component*** je koren svih GUI komponenata i kontejnera. Sve Swing GUI komponente (sem ***JFrame, JApplet i JDialog***) su podklase od klase ***JComponent*** . Na slici 1 se vide nejznačajniji metoda kojima se manipuliše svojstvima klasa ***Component, Container i JComponent***, kao što sz font, boja, kursor miša, veličina, alat za pirkaz tektsova i granice.



Slika 6.1 Sve Swing GUI komponehte nasležuju svojstva iy klasa Component, Container i JComponent

COMPONENT I CONTAINER - 1. DEO (VIDEO)

Ovo je prvi video od serije od 3 videa koji objašnjava razvoj GUI primenom Swing paketa klasa

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

COMPONENT I CONTAINER - 2. DEO (VIDEO)

Ovo je drugi video od serije od 3 videa koji objašnjava razvoj GUI primenom Swing paketa klasa

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

COMPONENT I CONTAINER - 3. DEO (VIDEO)

Ovo je treći video od serije od 3 videa koji objašnjava razvoj GUI primenom Swing paketa klasa

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

PRIMER 15 - KORIŠĆENJE ZAJEDNIČKIH SVOJSTAVA

Klasa LineBorder definiše graničnu liniju, a TitleBorder – naslov graničnog okvira.

Možete definisati granicu objekta klase JComponent, koristeći iskaz:

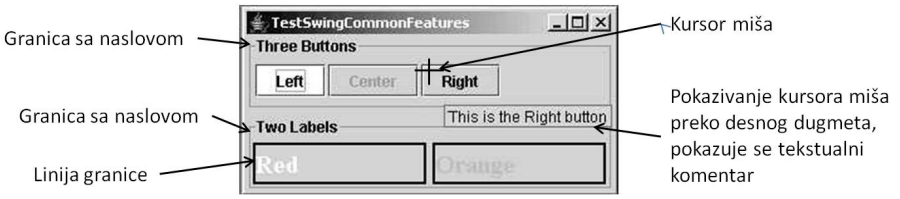
```
new LineBorder(Color color, int width);
```

gde je width debljina granične linije. Naslov granice se može definisati sa

```
new TitleBorder(String title);
```

Listing pokazuje zajednička svojstva Swing komponenata. Program kreira panel p1 koji sadrži tri dugmeta (linija 8) i panel p2 da nosi dve oznake (linija 26), kao što je pokazano na slici 2. Boja pozadine dugmeta ***jbtLeft*** je bela (linija 12) a boja prednjeg prikaza dugmeta ***jbtCenter*** je zelena (linija 13). Objašnjenje ta dugme jbtRight je dato u liniji 14. Naslovljena granica i za panele p1 i p2 (linija 18,37) a linije na granicama sa postavljeni na oznake (linije 33-34). Kursor miša je definisan sa krstiće u p1 (linije 19). Klasa Cursor sadrži konstante

sa specificiranje oblika kursora, kao što su; DEFAULT_CURSOR( ), CROSSHAIR_CURSOR ( ), HAND_CURSOR( ), MOVE_CURSOR( ), TEXT_CURSOR( ), i dr. Kreiran kursor sa ukrštenim linijama je formiran sledećim iskazom: ***newCursor(Cursor.CROSSHAIR_CURSOR). Unapred definisan oblik kursora se i dalje koristi za panel p2, jer nije definisan poseban oblik kursora za njegov slučaj. Isto svojstvo može imati druge početne vrednosti u različitim komponentama***



Slika 6.2 Kreiran panel sa GUI komponentama

```
1 import java.awt.*;
 2 import javax.swing.*;
 3 import javax.swing.border.*;
 4
 5 public class TestSwingCommonFeatures extends JFrame {
 6   public TestSwingCommonFeatures() {
 7  // Kreiranje panela  za grupu od tri dugmad
 8  JPanel p1 = new JPanel(new FlowLayout(FlowLayout.LEFT, 2, 2));
 9  JButton jbtLeft = new JButton("Left");
10  JButton jbtCenter = new JButton("Center");
11  JButton jbtRight = new JButton("Right");
12  jbtLeft.setBackground(Color.WHITE);
13  jbtCenter.setForeground(Color.GREEN);
14  jbtRight.setToolTipText("This is the Right button");
15  p1.add(jbtLeft);
16  p1.add(jbtCenter);
17  p1.add(jbtRight);
18  p1.setBorder(new TitledBorder("Three Buttons"));
19  p1.setCursor(new Cursor(Cursor.CROSSHAIR_CURSOR));
20
21  // Kreiranje fonta i linije granice
22  Font largeFont = new Font("TimesRoman", Font.BOLD, 20);
23  Border lineBorder = new LineBorder(Color.BLACK, 2);
24
25  // Kreiranje panela za grupo od dve oznake
26  JPanel p2 = new JPanel(new GridLayout(1, 2, 5, 5));
27  JLabel jlblRed = new JLabel("Red");
28  JLabel jlblOrange = new JLabel("Orange");
29  jlblRed.setForeground(Color.RED);
30  jlblOrange.setForeground(Color.ORANGE);
31  jlblRed.setFont(largeFont);
32  jlblOrange.setFont(largeFont);
33  jlblRed.setBorder(lineBorder);
34  jlblOrange.setBorder(lineBorder);
35  p2.add(jlblRed);
36  p2.add(jlblOrange);
37  p2.setBorder(new TitledBorder("Two Labels"));
```

```
38
39  // Dodavanje dva panela u okvir
40  setLayout(new GridLayout(2, 1, 5, 5));
41  add(p1);
42  add(p2);
43   }
44
45   public static void main(String[] args) {
46  //  Kreiranje okvira i postavljanje njegovih svojstava
47  JFrame frame = new TestSwingCommonFeatures();
48  frame.setTitle("TestSwingCommonFeatures");
49  frame.setSize(300, 150);
50  frame.setLocationRelativeTo(null); // Centar okvira
51  frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
52  frame.setVisible(true);
53   }
54 }
```

ZADACI 15 - 18

Samostalno vežbanje rada sa zajedničkim svojstvima Swing GUI komponenata.

Zadatak 15 - Kako postavljate boju pozadine, boju prednjeg prikaza, font, i alat za tekstualne komentare kod neke Swing komponente?

Zadatak 16 - Zašto alat za za tekstualne komentare se ne prikazuje u sledećem programu

```
1 import javax.swing.*;
 2
 3 public class Test extends JFrame {
 4   private JButton jbtOK = new JButton("OK");
 5
 6   public static void main(String[] args) {
 7  // Create a frame and set its properties
 8  JFrame frame = new Test();
 9  frame.setTitle("Logic Error");
10  frame.setSize(200, 100);
11  frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
12  frame.setVisible(true);
13   }
14
15   public Test() {
16  jbtOK.setToolTipText("This is a button");
17  add(new JButton("OK"));
18   }
19 }
```

Zadatak 17 - Prikaži rezultat izvršenja sledećeg programa:

```
import javax.swing.*;
public class Test {
  public static void main(String[] args) {
    JButton jbtOK = new JButton("OK");
    System.out.println(jbtOK.isVisible());

    JFrame frame = new JFrame();
    System.out.println(frame.isVisible());
  }
}
```

Zadatak 18. Šta će se desiti ako dodate dugme u kontejner nekoliko puta, kao što je pokazano dole? Da li to pravi sintaksnu grešku? Da li to pravi grešku izvršenja?

```
JButton jbt = new JButton();
JPanel panel = new JPanel();
panel.add(jbt);
panel.add(jbt);
panel.add(jbt);
}
```

Poglavlje 7

Ikone slika

IKONE

Ikona je slika fiksne veličine koja se može prikazati kod mnogih GUI komponenata. Ikone slika su objekti kreirani upotrebom klase ImageIcon.

Ikona je slika fiksne veličine. Obično je mala i koristi se sa dekoraciju komponenti. Slike se memorišu kao datoteke sa slikama (engl. image files). Java trenutno podržava tri formata: ***GIF (Graphic Interchange Format), JPEG (Joint Photographic Experts Group)*** i ***PNG*** (Portable Network Graphics). Ako imate slike u drugim formatima, morate da ih prvo konvertujete u jedan od ovih formnata, a pre upotrebe za dekoraciju GUI komponenti.

Da bi se prikazala ikona sa slikom, mora se najpre kreirati objekat klase ImageIcon upotrebom iskaza:***new javax.swing.ImageIcon(filename).*** Na primer, sledeći iskaz kreira ikonu iz slike **us.gif** iz direktorijama image:

```
ImageIcon icon = new ImageIcon("image/us.gif");
```

PRIMER 16 - RAD SA IMAGEICON

Ikona slike se može prikazati na mestu oznake (engl. label) ili dugmeta.

Ikona slike se može prikazati na mesto oznake (engl. label) ili dugmeta upotrebom iskaza: n***ew JLabel(imageIcon)*** ili ***new JButton(imageIcon).*** Listing klase **TestImageIcon** pokazuje prikazivanje ikona u oznakama i dugmadima. Primer kreira dva oznake i dva dugmeta sa ikonama, kao što je prikazano na slici 3. Ista granica i ikona slike se može koristiti kod više GUI komponenti

Slika 7.1 Ikone slika se pokazuju na natpisaima i dugamdima

Prilikom testiranja ovog primera možete koristiti bilo koje 4 slike. U tom slučaju vodite računa da u linijama koda 5-8 koristite njihove nazive.

```
1 import javax.swing.*;
 2 import java.awt.*;
 3
 4 public class TestImageIcon extends JFrame {
 5   private ImageIcon usIcon = new ImageIcon("image/us.gif");
 6   private ImageIcon myIcon = new ImageIcon("image/my.jpg");
 7   private ImageIcon frIcon = new ImageIcon("image/fr.gif");
 8   private ImageIcon ukIcon = new ImageIcon("image/uk.gif");
 9
10   public TestImageIcon() {
11  setLayout(new GridLayout(1, 4, 5, 5));
12  add(new JLabel(usIcon));
13  add(new JLabel(myIcon));
14  add(new JButton(frIcon));
15  add(new JButton(ukIcon));
16   }
17
18   /** Main metod */
19   public static void main(String[] args) {
20  TestImageIcon frame = new TestImageIcon();
21  frame.setTitle("TestImageIcon");
22  frame.setSize(200, 200);
23  frame.setLocationRelativeTo(null); // Center the frame
24  frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25  frame.setVisible(true);
26   }
27 }
```

DODAVANJE SLIKA I FOTOGRAFIJA U GUI (VIDEO)

Video objašnjava kako da ubacite neku sliku u vaš GUI

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

ZADACI 19 - 21

Samostalno vežbanje rada sa ImageIcon objektima.

Zadatak 19. Kako kreirate ImageIcon iz datoteke image/us.gif u direktorijumu klase?

Zadatak 20. Da li će sledeći program da prikaže tri dugmeta? Da li dugmad prikazuju istu ikonu?

```
import javax.swing.*;
import java.awt.*;

public class Test extends JFrame {

    public static void main(String[] args) {
        // Create a frame and set its properties
        JFrame frame = new Test();
        frame.setTitle("ButtonIcons");
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public Test() {
        ImageIcon usIcon = new ImageIcon("image/us.gif");
        JButton jbtn1 = new JButton(usIcon);
        JButton jbtn2 = new JButton(usIcon);

        JPanel p1 = new JPanel();
        p1.add(jbtn1);
JPanel p2 = new JPanel();
        p2.add(jbtn2);

        JPanel p3 = new JPanel();
        p2.add(jbtn1);

        add(p1, BorderLayout.NORTH);
        add(p2, BorderLayout.SOUTH);
        add(p3, BorderLayout.CENTER);
    }
}
```

Zadatak 21. Da li granica ili ikona može da se koristi od strane više GUI komponenata

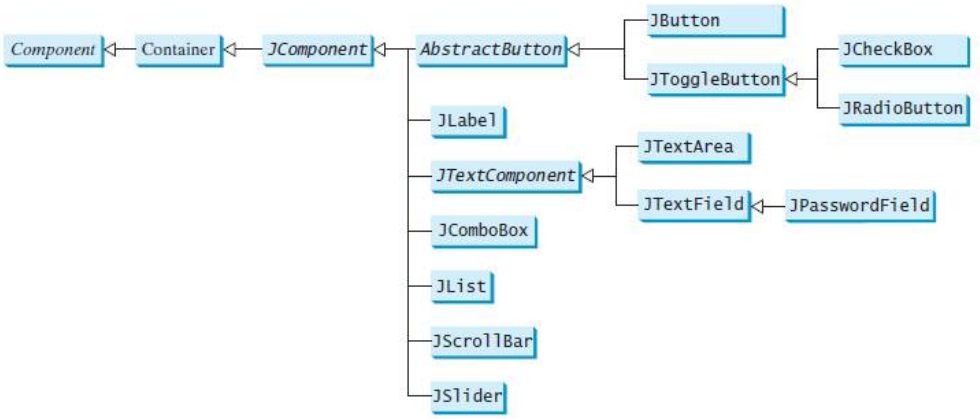
Poglavlje 8

Klasa JButton

SWING GUI KOMPONENTE

Da bi se kreiralo dugme interfejsa, koristi se klasa JButton.

Da bi koristili JButton u primerima demonstracije osnova GUI programiranja. Ovde ćemo prikazati nova svojstva klase ***JButton***. Slika 1 prikazuje Swing GUI komponente koje se najčešće koriste pri kreiranju korisničkih interfejsa. Jedna od tih GUI komponenta je ***JButton***.



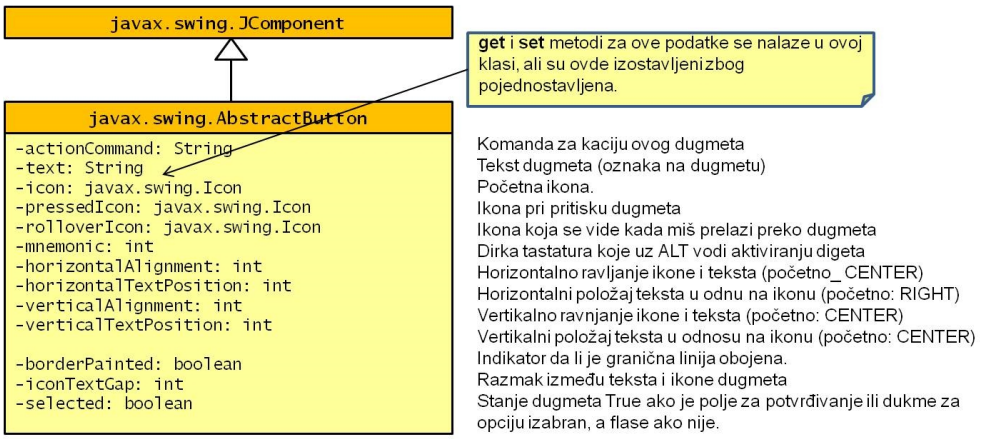
Slika 8.1 Swing GUI komponente

JBUTTON

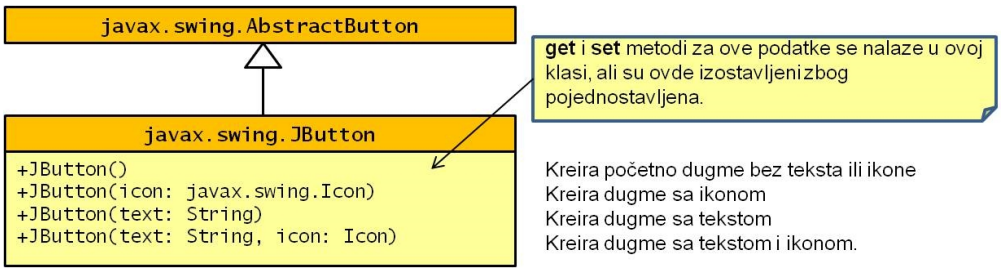
JButton kreira dugme GUI komponentu, koja kada se se klikne, pokreće određenu akciju.

Dugme je komponenta koja okida jednu akciju kada se klikne. Swing obezbeđiuje regularnu dugmad, dugmad za isključenje, dugmad polja za potvrđivanje, i dugme opicije (engl., radio button). Zajednička svojstva ovih dugmadi su definisana u ***javax.swing.AbstractButton***, kao što je prikazano na slici 2 .

JButton nasleđuje ***AbstractButton*** i obezbeđuje nekoliko konstruktora za kreiranje dugmad, kao što je pokazano na slici 3 .



Slika 8.2 Klasa AbstractButton definiše yajednička svojstva različitih tipova dugmadi



Slika 8.3 Konstruktor klase JButton

PRIMER 17 - KORIŠĆENJE KLASE JBUTTON

Ikona slike u vidu zastave nanosi se na površinu svakog od tri dugmeta.

Dugme ima početnu ikonu, ikonu pritisnutog dugmeta i ikonu dugmeta pri prelasku kursora miša preko njega.

Listing klase **TestButtonIcon** prikazuje zastavu SAD kao regularnu ikonu. Kanadsku – kao ikonu pritisnutog dugmeta, i zastavu Velike Britanije – ikona prelaska kursora miša preko dugmeta. (slika 4).

Slika 8.4 Primer tri dugmeta sa ikonim zastava

```
import javax.swing.*;

public class TestButtonIcons extends JFrame {

    public static void main(String[] args) {
```

```
        // Kreira okvir  i postavlja njegova svojstva
        JFrame frame = new TestButtonIcons();
        frame.setTitle("ButtonIcons");
        frame.setSize(200, 100);
        frame.setLocationRelativeTo(null); // Centar okvira
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public TestButtonIcons() {
        ImageIcon usIcon = new ImageIcon("image/usIcon.gif");
        ImageIcon caIcon = new ImageIcon("image/caIcon.gif");
        ImageIcon ukIcon = new ImageIcon("image/ukIcon.gif");

        JButton jbt = new JButton("Click it", usIcon);
        jbt.setPressedIcon(caIcon);
        jbt.setRolloverIcon(ukIcon);

        add(jbt);
    }
}
```

HORIZONTALNO I VERTIKALNO RAVNJANJE IKONE I TEKSTA DUGMETA

Metodi setHorizontalAlignment(int) i setVerticalAlignment(int) vršie se horizontalno, odn. Vertikalno ravnjanje ikone i teksta dugmeta

Horizontalno ravnjanje (slika 5) definiše kako se tekst i ikona postavljaju horizontalno na dugmetu. To se definiše metodom ***setHorizontalAlignment(int)*** sa jednom od pet konstanti***: LEADING, LEFT, CENTER, RIGHT***, ili ***TRAILING***, kao što je pokazano na slici 5. ***Trenutno LEADING i LEFT, kao i TRAILING i RIGHT*** su istovetne. Početno stanje je: AbstractButton.***CENTER.***

Vertikalno ravnjanje određuje kako se tekst i ikona vertikalno postavljaju na dugmetu (slika 6). To se postiže pozivom metoda setVerticalAlignment(int) sa jednom od tri konstante: TOP, CENTER ili BOTTOM , kao što je pokazano na slici 32. Početno, unapred definisano stanje je AbstractButton.CENTAR

Horizontalno levo | Horizontalno centar | Horizontalno desno

Slika 8.5 Horiyontalno ravnjanje ikona i teksta na dugmentu

Vertikalni vrh Vertikalnicentar Vertikalno dno

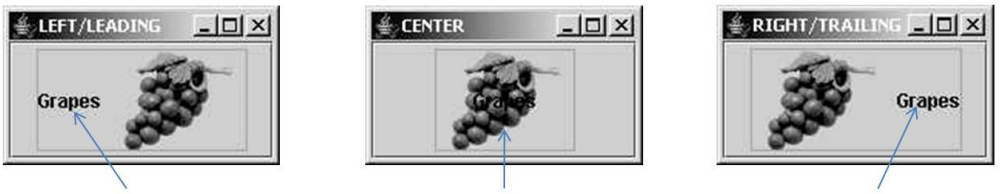
Slika 8.6 Vertikalno ravnjanje ikone i teksta na dugmetu

DEFINISANJE POLOŽAJA TEKSTA

Metodi setHorizontalTextPosition(int) i setVerticalTextPosition(int) određuju horizontalni ili vertikalni položaj teksta u odnosu na ikonu.

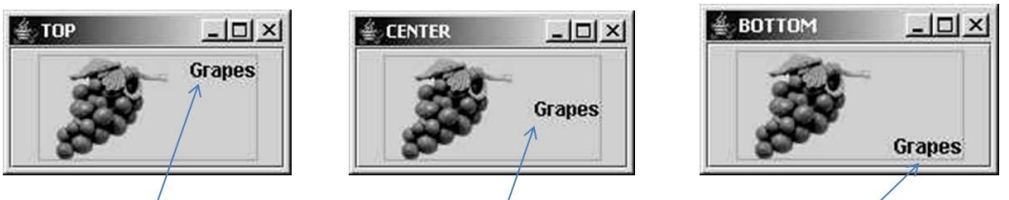
Određivanje položaja teksta u horizontali u odnosu na ikonu. To se vrši pozivom metoda ***setHorizontalTextPosition(int)*** sa jednom od pet konstanti***: LEADING, LEFT, CENTER, RIGHT, ili TRAILIN***, s tim što je za sada, ***LEADING i LEFT, kao i TRAILING i RIGHT*** daju isti efekat. Na slici 7 prikazama su tri karakteristična položaja.

Položaj teksta u vertikalnom pravcu se određuje u odnosu na ikonu. Položaj se određuje metodom ***setVerticalTextPosition(int)*** sa jednom od tri konstante ***TOP, CENTER ili BOTTOM***, kao što je prikazano na slici 8. Početni, unapred definisani položaj je ***AbastractButton.CENTER.***



Levi položaj teksta Položaj teksta u centru Desni položaj teksta

Slika 8.7 Horiyontalno postavljanje teksta u odnosu na ikonu



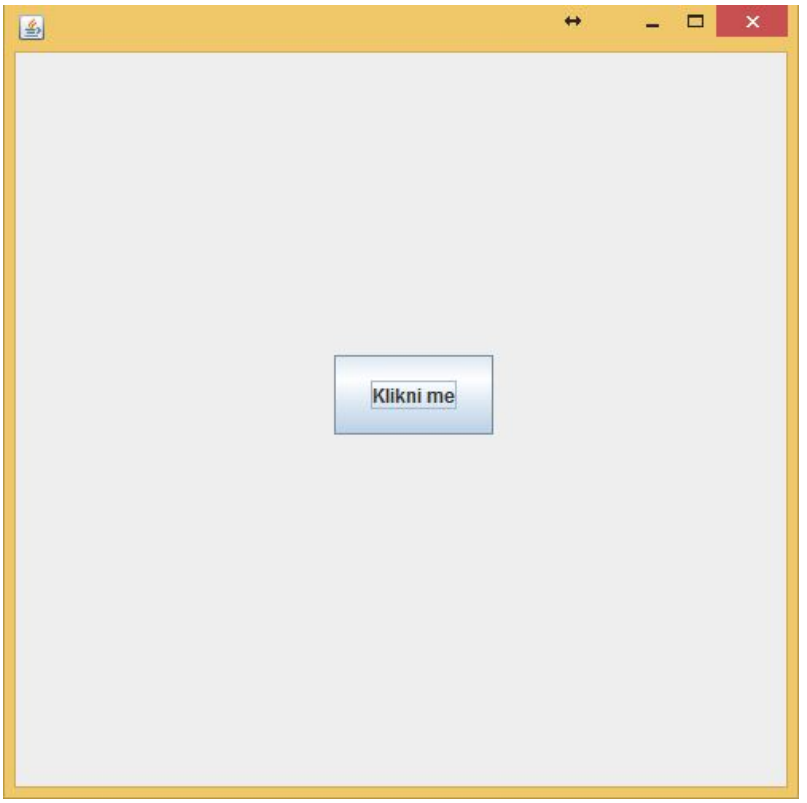
Položaj teksta na vrhu Položaj teksta u centru Položaj teksta na dnu

Slika 8.8 Vertikalno postavljanje teksta u odnosu na ikonu

PRIMER 18

Cilj ovog zadatka je prikazati način pravljenja dugmeta na formi u Javi.

Kreirati program na osnovu slike. Klikom na dugmepomoću JOptionPane-a, prikazati porukuKliknuli ste na dugme.Zadatak treba da bude deo projekta KI105-V11 i paketa zadatak7. Pokretačka klasa nosi naziv Zadatak7.



Slika 8.9 Zadatak 7

Objašnjenje:

JPanel nam omogućava pravljenje sadržaja unutar jednog prozora (JFrame).

ActionListener nam omogućava pravljenje događaja na klik dugmeta (JButton) ili neke druge Java GUI komponente.

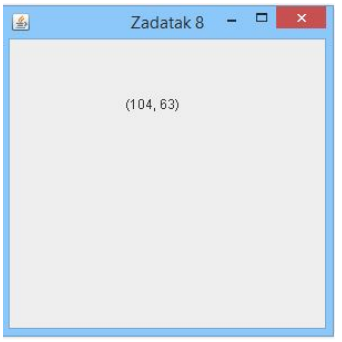
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package zadatak7;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

/**
 *
 * @author razvoj
```

```
 */
public class Zadatak7 extends JFrame {

    //Kreiramo panel na kome ce se nalaziti dugme
    JPanel panel = new JPanel();
    JButton btn = new JButton("Klikni me");

    public static void main(String[] args) {
        new Zadatak7();
    }

    public Zadatak7() {
        setSize(500, 500);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
        panel.setLayout(null);
        //Setujemo panel kao panel forme.
        setContentPane(panel);
        //Zovemo metodu koju smo napravili da bi odvojili kod
        setupButtons();
    }

    public void setupButtons() {
        //Konfiguracija dugmeta
        btn.setBounds(200, 190, 100, 50);
        btn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(rootPane, "Kliknuli ste dugme");
            }
        });
        //Dodajemo dugme na panel
        panel.add(btn);
    }
}
```

PRIMER 19

Cilj ovog zadatka je prikaz mogućnosti korišćenja kordinatnog sistema u Javi.

Napraviti program koji će klikom na formu u toj tački ispisati kordinate x i y. Zadatak treba da bude deo projekta KI105-V11 i paketa zadatak8. Pokretačka klasa nosi naziv Zadatak8.



Slika 8.10 Zadatak 8

Objašnjenje:

U panelu koji kreiramo u okviru klase PanelKoordinate definišemo da se na kreiranje objekta(u konstruktoru) postavlja mouseListener koji osluškuje klik na panel. U trenutku kada se klik desi, izvšava se kod u metodi mousePressed sa detaljima o događaju koji je sadržan u objektu MouseEvent. Iz tog objekta metodama getX i getY možemo uzeti tačne koordinate klika i te vrednosti postaviti kao koordinate objekta Point koji je atribut klase. Metodom repaint vršimo ponovno iscrtavanje panela, automatskim pozivom metode paintComponent koja prikazuje podatke atributa klase kome su prethodno promenjene koordinate.

Rešenje (Klasa Main):

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package zadatak8;

import javax.swing.JFrame;

/**
 *
 * @author razvoj
 */
public class Zadatak8 extends JFrame {

    public Zadatak8() {
        setTitle("Zadatak 8");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 300);
        setLocationRelativeTo(null);
        /**
         * Kao sadržaj prozora koristimo panel koji kreirano u posebnoj klasi
         */
        setContentPane(new PanelKoordinate());
        setVisible(true);
    }

    public static void main(String[] args) {
```

```
        new Zadatak8();
    }

}
```

Rešenje (Klasa PanelKoordinate):

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package zadatak8;

import java.awt.Graphics;
import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JPanel;

/**
 *
 * @author razvoj
 */
public class PanelKoordinate extends JPanel {

    private Point p = new Point(0, 0);

    public PanelKoordinate() {
        addMouseListener(new MouseAdapter() {
            @Override
            public void mousePressed(MouseEvent e) {
                if (e.getButton() == MouseEvent.BUTTON1) {
                    p.x = e.getX();
                    p.y = e.getY();
                    repaint();
                }
            }
        });
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawString("(" + p.x + ", " + p.y + ")", p.x, p.y);
    }
}
```

JLABEL, JBUTTON I JPANEL (VIDEO)

Vido objašnjava primenu klasa JLabel, JButton i JPanel Swing paketa klasa

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

ZADACI 22 - 24

Samostalno vežbanje upotrebe JButton objekata.

Zadatak 22 - Kako kreirate dugme sa tekstom OK? Kako menjate tekst na dugmetu? Kako postavljate ikonu, ikonu sa pritiskom, ikonu koja se pojavjjuje kada kursor prelazi preko dugmeta.

Zadatak 23 - Za objekat jbtOK klase ***JButton***, napišite iskaz da postavite crvenu boju u prednji prikaz, u žuto pozadinu, dirka za skraćeni pristup K, alat za tekst saveta „Click OK to proceed“, horizontalno ravnjanje LEFT, vertikalno pozicija teksta sa TOP, i razmak ikona je 5.

Zadatak 24 - Izlistaj najmanje pet svojstva definisanih u ***AbstractButton*** klasi.

Poglavlje 9

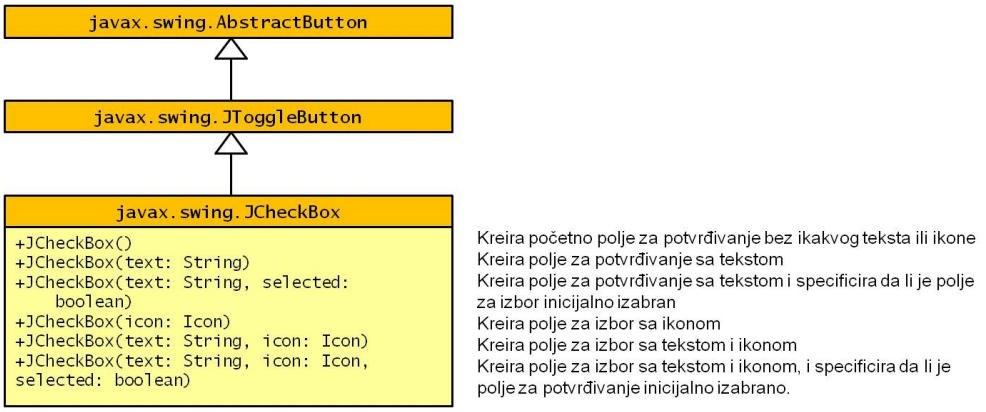
Klasa JCheckBox

JCHECKBOX

Za kreiranje polja za potvrđivanje (check box button) koristi se klasa JCheckBox

Preklopno dugme (engl. toggle button) ili dugme potvrđivanja ima dva stanja kao prekidač. Klasa***JToggleButton*** nasleđuje ***AbstractButton*** i primenjuje preklopno dugme. Često se podklase klase ***JToggleButton*** – klase ***JCheckBox*** i ***JRadioButton*** se koriste da bi se i korisniku omogućio izbor tipa da ili ne.

JCheckBox nasleđuje svojstva klase AbstractButton, kao što su: text, icon, mnemonic, ***verticalAlignment, horizontalAlignment, horizontalTextPosition, verticalTextPosition,*** i selected. Klasa ***CheckBox*** nudi nekoliko konstruktora za kreiranje polja za potvrđivanje, kao što je pokazano na slici 1 .



Slika 9.1 Klasa CheckBox obezbeđuje dugme za potvrđivanje

PRIMER 20 - KORIŠĆENJE JCHECKBOX KLASE

Konstruktor klase JCheckBox kreira objekat "Student" koji predstavlja dugme za potvrđivanje

Radi ilustracije, ovde se daje primer kreiranje polja za potvrđivanje sa tekstom Student. Boje prednjeg izgleda je crvena, pozadinska boja je bela, dirka skraćivanja (mnemonic) je S i inicijalno je izabrana:

```
JCheckBox jchk = new JCheckBox("Student", true);
jchk.setForeground(Color.RED);
```

```
jchk.setBackground(Color.WHITE);
jchk.setMnemonic('S');
```

Dugmetu se može pristupiti i korišćenog skraćenog puta, korišćenjem dirki tastature (mnemonics). Istovremenim pritiskom dirki tastature Alt+S postiže se isti efekat kao i korišćenjem polja za potvrživanja korisničkog interfejsa i miša. Metodom isSelected() vrši se provera da li je preklopno dugme polja za potvrđivanje bilo izabrano (potvrđeno) ili nije.

ZADATAK 25

Samostalno vežbanje upotrebe JCheckBox objekata.

ZADATAK ZA VEŽBU

Kako bi kreirali polje za potvrđivanje? Kako ga kreirate sa inicijalnom opcijom da je izabran (potvrđen)? Kako određujete da li je kontrolno dugme izabrano (potvrđeno)?

PRIMENA KLASE JCKECKBOX SA NETBEAN IDE

Vido objašnjava primenu klase JCheckBox sa NetBean IDE

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

Poglavlje 10

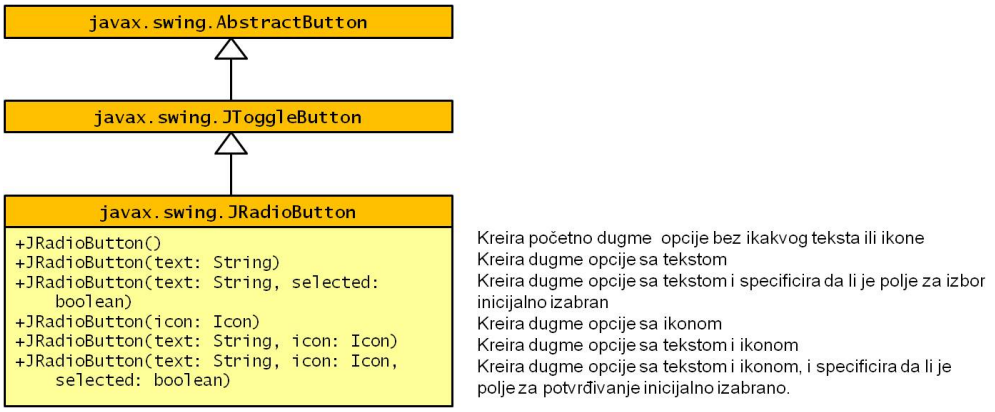
Klasa JRadioButton

JRADIOBUTTON

Za kreiranje dugmeta opcije, koristi se klasa JRadioButton.

Dugmad za opcije, ili radio dugmad, omogućavaju vam da izaberete jednu opciju iz grupe opcija. Pojavno, dugmad opcija podsećaju na dugmad potvrđivanja, ali dugmad potvrđivanja imaju oblik kvadrata (praznog ili overenog), a dugmad opcija imaju oblik kruga (praznog ili popunjenog). Ako je krug prazan, znači da opcija nije izabrana.

Klasa ***JRadioButton*** nasleđuje svojstva klase ***AbstractButton*** i obezbeđuje nekoliko konstruktora za kreiranje dugmad opcija (radio dugmad), prikazanih na slici 36. Ovi konstruktori su slični konstruktorima klase ***JCheckBox***



SLika-1: Klasa JRadioButton definiše dugme za opcije

PRIMER 21 - KORIŠĆENJE KLASE JRADIOBUTTON

Klasa JRadioBox kreira dugme za opcije, kao i boju naslova i boju pozadine.

Radi ilustracije, daje se primer kreiranja dugmeta opcije sa tekstom Student. Sledeći deo programa definiše crvenu boju prednjeg izgleda, belu boju pozadine, dirku za skraćen pristup (menumonic) S, koja ja inicijalno izabrana:

```
JRadioButton jrb = new JRadioButton("Student", true);
jrb.setForeground(Color.RED);
jrb.setBackground(Color.WHITE);
jrb.setMnemonic('S');
```

Da bi formiralu grupu dugmadi opcija, morate dormirati primerak klase ***java.swing.ButtonGroup*** i da upotrebite metod **add**, na sledeći način:

```
ButtonGroup group = new ButtonGroup();
group.add(jrb1);
group.add(jrb2);
```

Ovaj program kreira grupu sa dva dugmeta opcija: jrb1 i jrb2. Kako su u grupi, samo jedan od njih može d abude izabran. Da nisu u grupi, mogu se birati nezavisno, tj. mogu više od jednog da budu izabrani.

ZADATAK 26

Samostalno vežbanje upotrebe JRadioButton objekata.

ZADATAK ZA VEŽBU

Kako kreirate dugme opcije? Kako kreirate dugme opcije sa inicijalnim statusom izabranog? Kako grupišete dugmad izbora zajedno? Kako određujete da li je dugme izbora izabrano?

JRADIOBUTTON (VIDEO)

Video objašnjava primenu klase JRadioButton

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

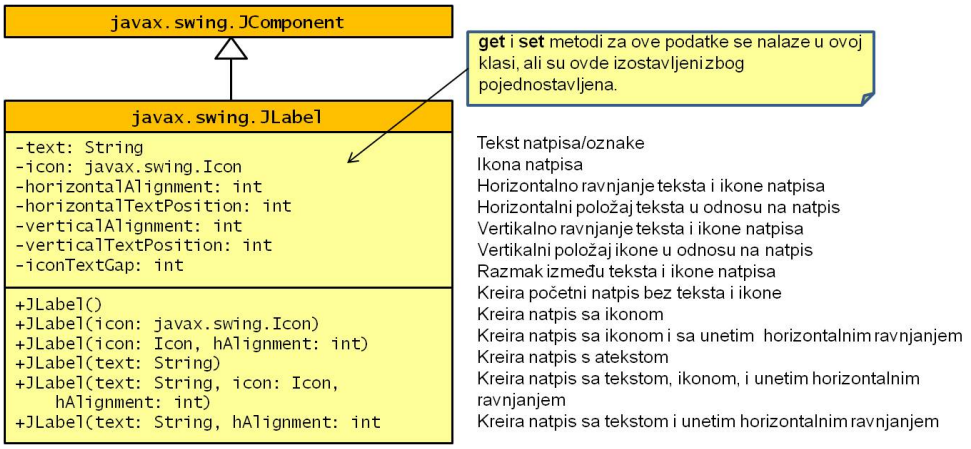
Poglavlje 11

Natpisi (JLabel)

KLASA JLABEL

Da bi se kreirao natpis(oznaka u korisničkom interfejsu), koristi se klasa JLabel

Natpis ili oznaka je prostor za prikazivanje kratkog teksta, slike ili i jednog i drugog. Upotrebljava se radi označavanja drugih komponenata korisničkog interfejsa (najčešće tekstualnih polja). Na slici 1 prikazan je klasni dijagram klase ***JLabel.***



SLika-1: Klasa JLabel prikazuje teks, ikoni ili jedno i drugo

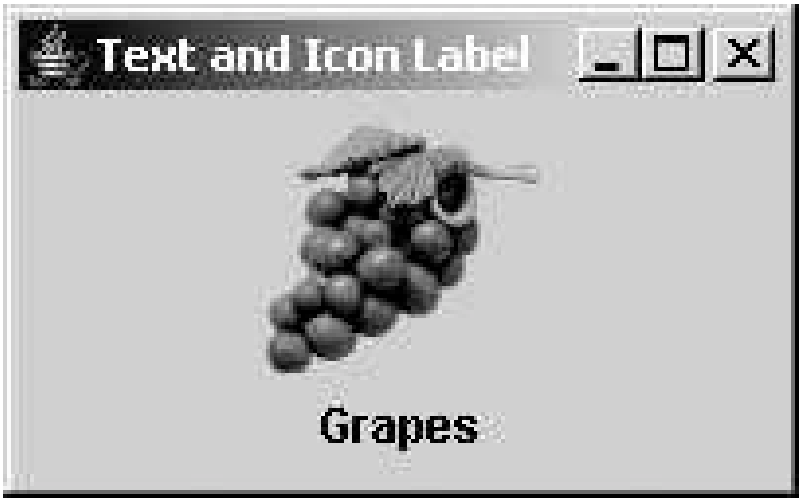
PRIMER 22 - PRIMENA KLASE JLABEL

Klasa JLabel postavlja natpis i ikonu

JLabel nasleđuje sva svojstva klase ***JComponent*** i ima mnogo svojstva slična sa klasom***Button***, kao što su***: text, icon, horizontalAlignment, verticalAlignment, horizontalTextPosition, verticalTextPosition***, i ***iconTextGap***. Na primer, sledeći program prikazuje natpis sa tekstom i ikonom (slika 2):

```
// Create an image icon from an image file
ImageIcon icon = new ImageIcon("image/grapes.gif");
// Create a label with a text, an icon,
// with centered horizontal alignment
JLabel jlbl = new JLabel("Grapes", icon, JLabel.CENTER);
//Set label's text alignment and gap between text and icon
jlbl.setHorizontalTextPosition(JLabel.CENTER);
```

```
jlbl.setVerticalTextPosition(JLabel.BOTTOM);
jlbl.setIconTextGap(5);
```

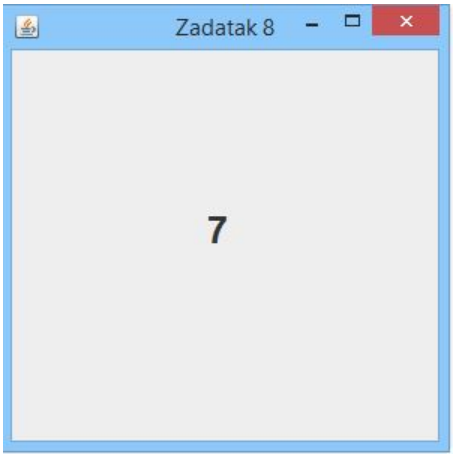


Slika 11.1 Prikaz JLabel objekta

PRIMER 23 - BROJAČ NA LABELI

Cilj ovog zadatka je prikaz korišćenja Timera u Javi.

Napraviti program koji će brojati sekunde i sekunde ispisavati na formi kao na slici. Zadatak treba da bude deo projekta KI105-V11 i paketa zadatak9. Pokretačka klasa nosi naziv Zadatak9.



Slika 11.2 Zadatak 9

Objašnjenje:

Timer nam omogućava da određenu operaciju ponovimo na n milisekundi. Konkretno u ovom zadatku stavljamo 1000 milisekundi što je jedna sekunda i u tajmeru pozivamo operaciju actionOnButton koja povećava trenutan broj za jedan i ispisuje ga u okviru komponente JLabel.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package zadatak9;

import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.Timer;

/**
 *
 * @author razvoj
 */
public class Zadatak9 extends JFrame {

    JLabel label = new JLabel("1");
    JPanel panel = new JPanel();

    public Zadatak9() {
        Timer t = new Timer(1000, new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                actionOnButton();
            }
        });
        t.start();
        setTitle("Zadatak 8");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 300);
        setLocationRelativeTo(null);
        setVisible(true);
        setContentPane(panel);
        panel.setLayout(null);
        label.setFont(new Font("Arial", 1, 25));
        panel.add(label);
        label.setBounds(130, 110, 30, 20);

    }

    public static void main(String[] args) {
        new Zadatak9();
    }

    public void actionOnButton() {
        int trenutanBr = Integer.parseInt(label.getText());
        trenutanBr++;
        label.setText(trenutanBr + "");
```

```
    }
}
```

ZADACI 27 I 28

Samostalno vežbanje upotrebe JLabel objekata.

ZADACI ZA VEŽBU

Zadatak 27. Kako kreirate natpis sa tekstom: Address? Kako menjate ime natpisa? Kako postavljate ikonu u naslov?

Zadatak 28. Za dati objekat klase JLabel, nazvanim jlbMap, napišite iskaze kojim bi odredili crvenu boju za prednji izgled natpisa, žutu boju za pozadinu, dirku za skraćeni pristup (menumonic) M, alata sa tekstualni savet Map image, horizontalno ravnjanje RIGHT, vertikalno ravnjanje BOTTOM, horizontalna pozicija teksta LEFT, vertikalni položaj teksta TOP, i razmak između teksta i ikone da bude 5.

JLABEL, JBUTTON I JPANEL (VIDEO)

Video daje primer korišćenja klasa JLabel, JButton i JPanel

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

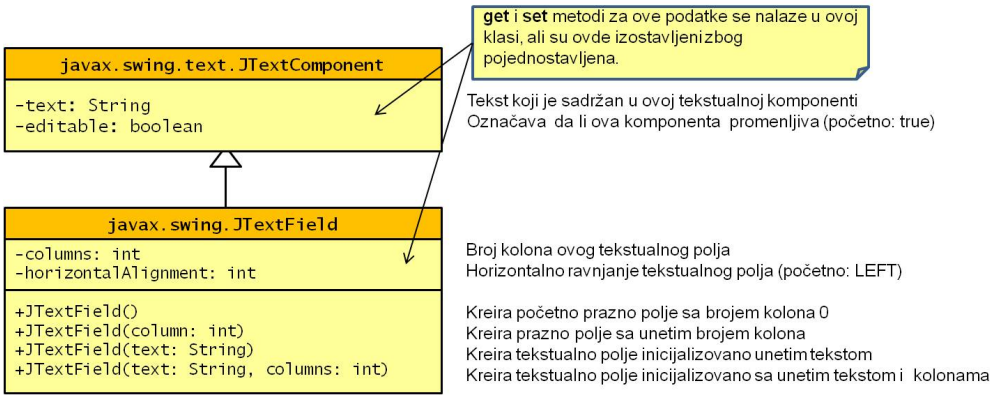
Poglavlje 12

Polje teksta (JTextField)

KLASA JTEXTFIELD

Da bi se kreiralo polje teksta u korisničkom interfejsu, koristi se klasa JTextField.

Polje teksta se korsti za unos ili prikaz nekog stringa (teksta). Klasa ***JTextField*** je podklasa klase ***JTextComponent.*** Slika 1 prikazuje klasni dijagram klase ***JTectField***.



Slika 12.1 Klasa JTextField pmogućava unos teksta

PRIMER 24 - KORIŠĆENJE KLASE JTEXTFIELD

Da bi se uneo novi tekst u tekstualno polje, upotrebljava se metod setText(newText).

Klasa ***JTextField*** nasleđuje klasu ***JTextComponent***, koja nasleđuje ***JComponent.*** Evo primera kreiranja tekstualnog polja sa crvenim prednjim izgledom i desnim horizontalnim ravnanjem:

```
JTextField jtfMessage = new JTextField("T-Storm");
jtfMessage.setForeground(Color.RED);
jtfMessage.setHorizontalAlignment(JTextField.RIGHT);
```

Da bi se uneo novi tekst u tekstualno polje, upotrebljava se metod setText(newText). Da bi se dobio tekst iz tekstualnog polja, koristi se metod getText().

Ako se tekstualno polje koristi za unos lozinke, upotrebite klasu JPaswordField umesto klase JTextField. Klasa JPasswordField je podklasa klasa JTextField i krije uneti tekst je se umesto unetih karaktera prikazuju zvezdice (*****).

JLABEL I JTEXTFIELD (VIDEO)

Video objašnjava primenu JLabel i JTextFiled

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

ZADATAK 29

Samostalno vežbanje upotrebe JTextField objekata.

ZADACI ZA VEŽBU:

1. Kako kreirate tekstualno polje sa 10 kolona i početnim tekstom: „Welcome to Java“? Kako pišete program za proveru da li je tekstualno polje prazbo?
2. Kako kreirate tekst polje za unos lozinki?

PRIMER SA JTEXTFIELD (VIDEO)

Video daje jedan primer primene JTextField klase

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

Poglavlje 13

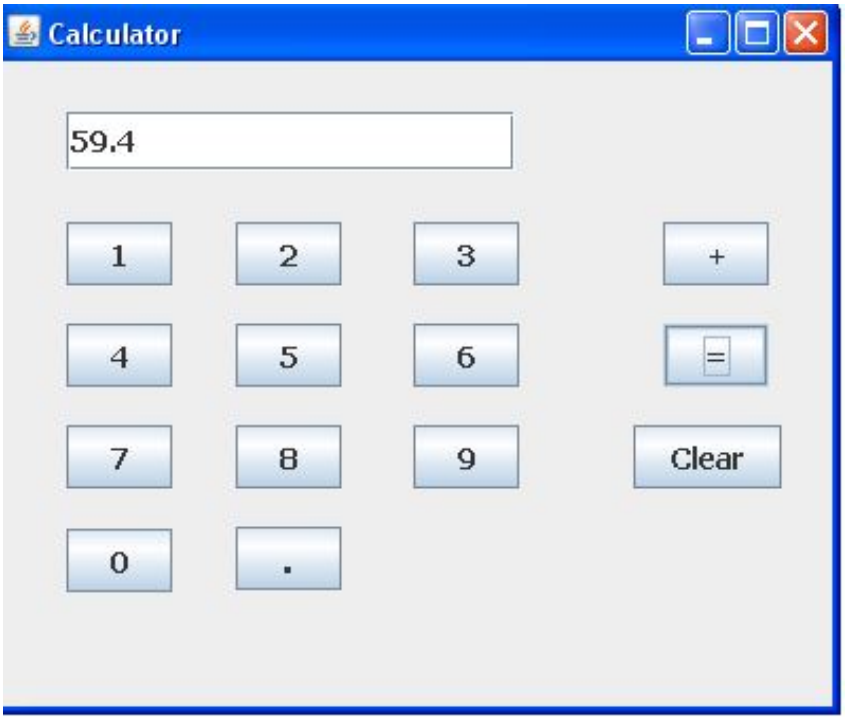
Domaći zadaci

ZADACI 1 I 2

Izrada programa na osnovu zadate slike interfejsa (zadatak 1 i 2)

Zadatak 1

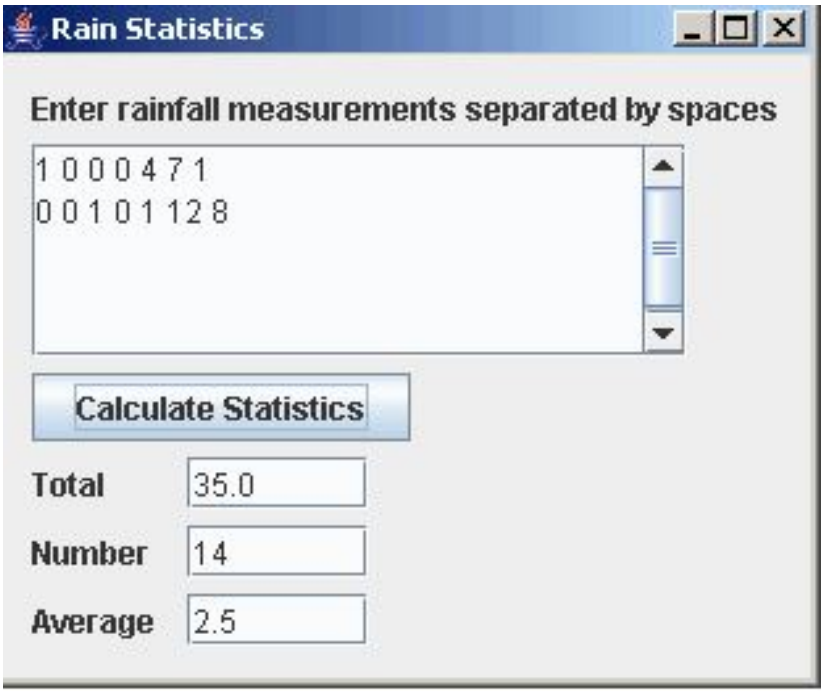
Napraviti GUI na osnovu sledeće slike:



Slika 13.1 Zadatak 1

Zadatak 2

Napraviti GUI na osnovu sledeće slike:



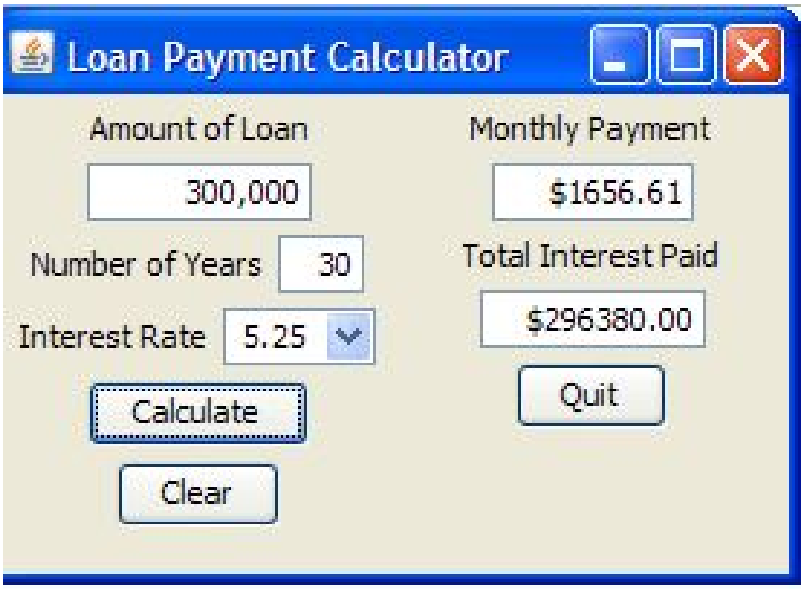
Slika 13.2 Zadatak 2

ZADACI 3 I 4

Izrada programa na osnovu zadate slike interfejsa (zadatak 3 i 4)

Zadatak 3

Napraviti GUI na osnovu sledeće slike:



Slika 13.3 Zadatak 3

Zadatak 4

Napraviti GUI na osnovu sledeće slike:

Slika 13.4 Zadatak 4

Zaključak

REZIME

Glavne pouke lekcije

1. Svaki kontejner ima svog menadžera rasporeda koji se koristi za određivanje položaja i mesta komponenti kontejnera u željenim lokacijama. Tri često korišćenih menadžera rasporeda su: ***FlowLayout, GridLayout, i BorderLayout***
2. Možete koristiti klasu ***JPanel*** kao podkontejner za grupisanje komponenti i postizanje željenog rasporeda.
3. Upotrebite add metod za postavljanje komponenti u ***JFrame*** ili ***JPanel***. Inicijalno, raspored okvira određen je sa BorderLayout, a raspored panela ***JPanel*** – sa FlowLayout.
4. Boje se definišu kombinacijom crvene, zelene i plave boje. Svaka od njih se definiše nijansom označenu brojem od 0 (najtamnija) do 255 (najsvetlija). To je tzv. RGB model.
5. Da bi se kreirao objekat ***Color***, koristi se ***new Color(r, g, b)*** u kome r, g i b definišu boju sa svojom crvenom, zelenom i plavom komponentom. Umesto toga. Možete koristiti 13 unapred definisanih, standardnih boja (***BLACK, BLUE, CYAN, DARK_GRAY, GRAY, GREEN, LIGHT_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW***) definisane kao konstante u klasi JComponent.
6. Svaka Swing GUI komponenta je podklasa klase javax.swing.Jcomponent, a ona je podklasa klase java.awt.Component. Zato, one nasleđuju svojstva, kao što su: ***font, background, foreground, height, width, i preferredSize*** iz klase ***Component***, toolTipText i border iz klase ***JComponent***.
7. Možete koristiti granice na svakoj Swing komponenti. Možete kreirati ikonu sa slikom upotrebom klase ImageIcon i prikazati je u nalepnici ili na dugmetu. Iste ikone i granice se mogu koristiti od više komponenti.
8. Možete prikazati tekst i ikonu na dugamdima (***JButton, JCheckBox, JRadioButton)*** i natpise (***JLabel***).
9. Možete definisati horizontalno i vertikalno ravnjanje teksta u ***JButton, JCheckBox, JRadioButton, i JLabel***, i horizontalno ravnjanje teksta u ***JTextField, kao i*** horizontalni i vertikalni položaj teksta relativno u odnosu na ikonu u ***JButton, JCheckBox, JRadioButton, i JLabel.***