

7.1. Layout Managers - Upravljači rasporeda

7.2. Dimenzije komponenti

7.3. Osnovni upravljači rasporedom

7.3.1. FlowLayout

7.3.2. BorderLayout

7.3.3. Komponenta JPanel

7.3.4. Komponenta JSplitPane

7.3.5. GridLayout

7.3.6. BoxLayout

7.4. Tekst komponente

7.5. Komponenta JScrollPane

7.1. Layout Managers - Upravljači rasporeda

U Javi postoje tri aspekta vezana za kreiranje kompleksnih Grafičkih Korisničkih Interfejsa (GUI):

- 1) Komponente koje čine ekran
- 2) Obrada događaja koja reaguju na interakciju sa komponentama
- 3) Pozicioniranje komponenti radi postizanja kvalitetnog rasporeda istih

Upravljanje rasporedom komponenti je neophodno jer je princip u Javi da svaki program koji je u njoj razvijen podjednako dobro radi na svakoj platformi. Komponente same po sebi, ne izgledaju isto na svakoj platformi niti zauzimaju podjednak prostor na ekranu. Obrada ovih principa upravljanja rasporedom je posao menadžera rasporeda (Layout Manager).

Layout management je mehanizam koji nam omogućava definisanje pravila prostornog rasporeda vizuelnih komponenti. Ovim se omogućava pravilan prikaz komponenti pri promeni geometrije kontejnera kao i konzistentan prikaz na uređajima različite rezolucije.

Svi upravljači moraju implementirati interfejs *LayoutManager* ili podinterfejs *LayoutManager2* koji definišu metode koje svaki upravljač mora da ima. Objekat klase koja opisuje upravljača (što znači da implementira interfejs *LayoutManager*) se pridružuje kontejneru metodom *setLayout*. Ova metoda je implementirana u klasi *java.awt.Container*. Dakle, za svaki kontejner u GUI aplikaciji, biramo jedan od postojećih menadžera, postavljamo ga kao upravljač kontejnera pozivom metode *setLayout* čime se raspoređivanje komponenti u kontejneru prepušta izabranom menadžeru rasporeda.

7.2. Dimenzije komponenti

Za svaku komponentu Swinga mogu biti zadate tri veličine komponente. Ove veličine samo po sebi ništa ne znače, tek u kombinaciji sa izabranim upravljačem rasporeda (LayoutManager-om) biće određeno koju od te tri veličine će komponenta imati.

Veličina komponente	Metode klase JComponent
željena (preferred)	public void setPreferredSize(Dimension preferredSize)
	public Dimension getPreferredSize()
minimalna (minimum)	public void setMinimumSize(Dimension minimumSize)
	public Dimension getMinimumSize()
maksimalna (maximum)	public void setMaximumSize(Dimension maximumSize)
	public Dimension getMaximumSize()

Sve tri veličine služe menadžerima rasporeda da podese dimenzije komponenti tako da posle svake operacije promene dimenzija prozora (kontejnera) u kojem se one nalaze, njihova veličina ostane odgovarajuća.

Upravljači rasporedom mogu ali nisu obavezni da poštuju preferirane, minimalne i maksimalne veličine komponenti koje raspoređuju. Pri korišćenju upravljača potrebno je u dokumentaciji videti koje dimenzije upravljač poštuje. Dakle ukoliko se koriste upravljači rasporeda, postavljene dimenzije komponenti se menjaju u zavisnosti od izbora upravljača.

Klasa `java.awt.Dimension` može se shvatiti kao pomoćna klasa koja se koristi u radu sa Swing komponentama i upravljačima rasporeda i služi za predstavljanje dimenzija komponenti. Ona enkapsulira dužinu i visinu komponente koju predstavlja kao 2 celobrojna, nenegativna atributa (atributi su tipa `integer`): `width` i `height`.

Osnovne metode klase `Dimension` su:

Metoda	Opis
<code>Dimension()</code>	Konstruktor, kreira novu instancu klase <code>Dimension</code> bez inicijalnog određivanja dužine i visine objekta.
<code>Dimension(int width, int height)</code>	Konstruktor, kreira novu instancu klase <code>Dimension</code> , istovremeno određuje vrednosti dužine i visine objekta.
<code>double getHeight()</code>	Vraća vrednost atributa <code>height</code>
<code>double getWidth()</code>	Vraća vrednost atributa <code>width</code>
<code>void setSize(Dimension d)</code>	Metoda postavlja nove vrednosti atributa <code>height</code> i <code>width</code> koje uzima iz prosleđenog objekta tipa <code>Dimension</code> .
<code>void setSize(int width, int height)</code>	Metoda postavlja nove vrednosti atributa <code>height</code> i <code>width</code> koje uzima iz prosleđenih parametara.

7.3. Osnovni upravljači rasporedom

Postoji osam osnovnih stilova rasporeda komponenti (na vežbama će biti korišćeni oni koji su označeni zelenom bojom):

1. **FlowLayout**
2. **BorderLayout**
3. **CardLayout**
4. **GridLayout**
5. **GroupLayout**
6. **BoxLayout**
7. **SpringLayout**
8. **GridBagLayout**

Svaki od navedenih upravljača rasporeda na drugačiji način raspoređuje komponente koje se nalaze u nekom od kontejnera (uglavnom ćemo za kontejnere koristiti JFrame, JPanel i JDialog). Rad sa upravljačima se svodi na prepoznavanje pogodnog upravljača i njegova dodela kontejneru u koji će se smeštati komponente. Svaki objekat tipa `java.awt.Container` poseduje metodu `setLayout(LayoutManager layoutManager)` sa kojom se može dodeliti željeni upravljač. Svaki kontejner poseduje i neki default `LayoutManager` po kome raspoređuje komponente unutar sebe ukoliko on nije promenjen pozivanjem navedene metode.

7.3.1. FlowLayout

Najjednostavniji menadžer rasporeda. Raspoređuje GUI komponente na način toka (flow) koji liči na pisanje sa leva na desno. Kada se dođe do kraja reda koji je određen širinom kontejnera, komponente prelaze u sledeći red. Flow layout manager slaže komponente poštujući njihovu preferiranu (željenu) veličinu.



FlowLayout je default menadžer za svaku JPanel komponentu. Ukoliko se FlowLayout menadžer kreira pomoću konstruktora bez argumenata poravnanje komponenti koje smeštaju u kontejner biće centralno:

```
new FlowLayout()
```

Primer 1. Importovati u Eclipse projekat Termin7i8 sa materijala i pokrenuti projekat sa ulaznim argumentom „1a“. Pogledati i prokomentarisati kod klase FlowLayoutFrame, obratiti pažnju na dodelu layout managera JFrame-u. Pauza koja se javlja između dodavanja komponenti u kontejner isključivo služi da se uvidi način na koji se raspoređuju komponente.

FlowLayout može biti prilagođen zadavanjem stila poravnavanja komponenti unutar kontejnera:

```
new FlowLayout (int align)
```

Parametar poravnanja može da bude jedan od ovih pet konstanti klase `FlowLayout` (vrednosti za parametar *align* konstruktura):

- a) `FlowLayout.CENTER`
- b) `FlowLayout.LEFT`
- c) `FlowLayout.RIGHT`
- d) `FlowLayout.LEADING`
- e) `FlowLayout.TRAILING`

Ukoliko se ne dodeli vrednost ovog parametra u konstruktoru podrazumevana vrednost je `FlowLayout.CENTER`. Konstante *LEADING* i *TRAILING* se koriste za bolju obradu orijentacije jezički prihvatljivih komponenti različitih od onih sleva nadesno.

(pokrenuti Termin 7 i 8 u Eclipsu sa ulaznim parametrom 1b)

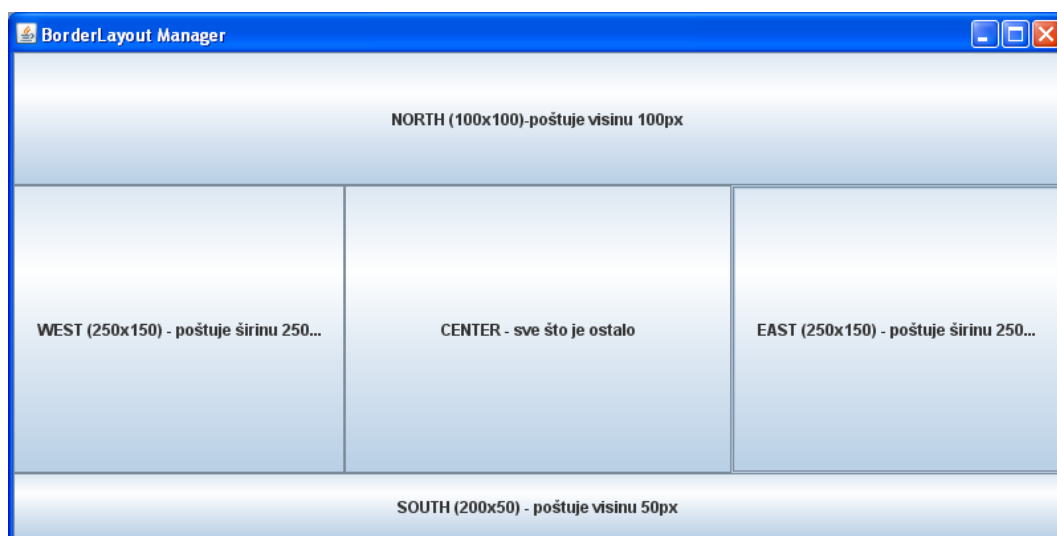
Postoji i treći `FlowLayout` konstruktor koji omogućava postavljanje razmaka između komponenti koje se dodaju u kontejner u pikselima u horizontalnom i vertikalnom smeru:

```
new FlowLayout(int align, int hgap, int vgap)
```

(pokrenuti Termin 7 i 8 u Eclipsu sa ulaznim parametrom 1c)

7.3.2. BorderLayout

Ovaj upravljač slaže komponente po stranama sveta (NORTH, SOUTH, EAST, WEST i CENTER). Raspoređivanje se obavlja tako što se prvo podesi veličina severne i južne komponente tako da se uzme preferirana (željena) visina komponente a za širinu se uzima širina kontejnera. Zatim se postavlja veličina istočne i zapadne komponente tako što se uzme njihova preferirana širina a za visinu se uzima preostao prostor i na kraju se sav preostao prostor dodeljuje centralnoj komponenti.



BorderLayout je osnovni menadžer rasporeda za Frame, JFrame kao i za JApplet. Prilikom dodavanja komponente u kontejner koja za menadžer rasporeda ima BorderLayout, metoda *add* ima dva parametra:

add (Component component, Object constraints)

Dakle prosleđujemo komponentu koju želimo dodati u kontejner ali biramo i jednu od 5 pozicija na kontejneru u koju želimo staviti komponentu. Drugi parametar u pozivu metode *add* može imati jednu od sledećih konstanti:

1. BorderLayout.NORTH
2. BorderLayout.SOUTH
3. BorderLayout.EAST
4. BorderLayout.WEST
5. BorderLayout.CENTER

Pokrenuti Termin7i8 u Eclipsu sa ulaznim parametrom 2a. Pogledati i prokomentarisati kod klase FlowLayoutFrame.

Kada se koristi BorderLayout menadžer niste u obavezi da zadate svih pet komponenti. Ukoliko na primer nije zadata središnje (CENTER) komponenta ivične komponente (NORTH, SOUTH, EAST, WEST) ne rastu da bi popunile prazninu koja je nastala zbog nedostatka središnje komponente. Samo ako je izostavljena jedna ili više ivičnih komponenti, prostor koje je ona trebala da popuni biće zauzet drugim komponentama.

Kao kod FlowLayout Manager-a postoji konstruktor koji zadaje dodatan horizontalan i/ili vertikalni razmak između komponenti u kontejneru.

Pokrenuti Termin7i8 u Eclipsu sa ulaznim parametrom 2b. Pogledati i prokomentarisati kod klase BorderLayoutFrame.

7.3.3. Komponenta JPanel

Klasa JPanel nudi klasu kontejnera niže kategorije u odnosu na „Top level“ kontejnere. Da bi se grupisale komponente u celinu kada se uređuje ekran može se koristiti i klasa JPanel.

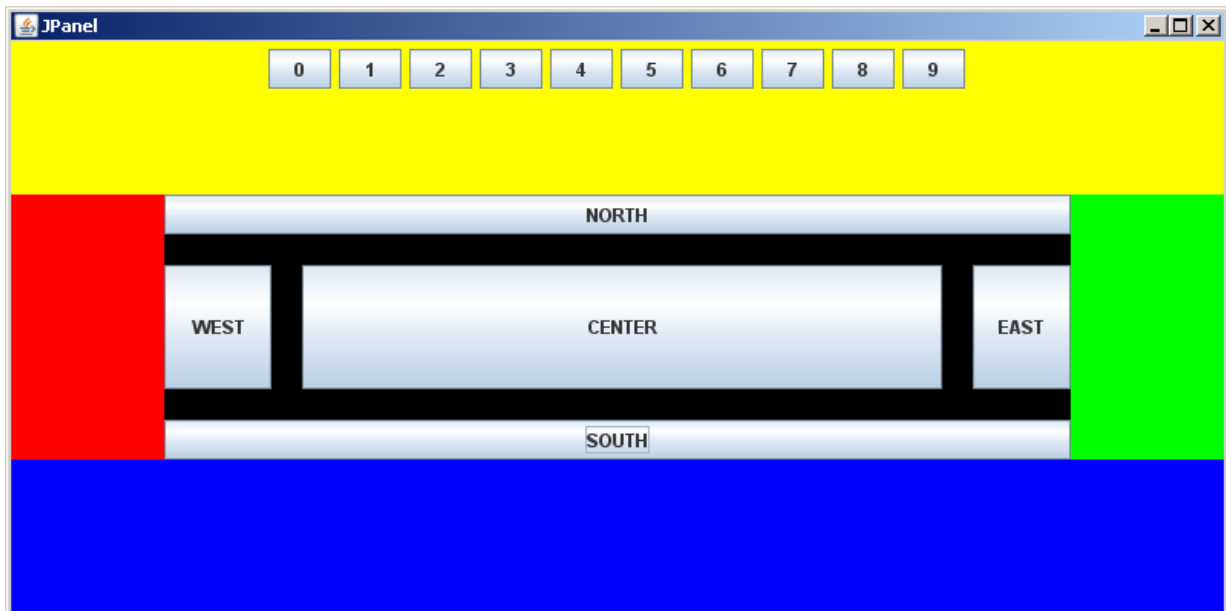
FlowLayout je default menadžer za JPanel komponentu, međutim slično kao i kod kontejnera može se postaviti proizvoljan LayoutManager za klasu JPanel.

Da bi se dodala komponenta u panel koristi se standardno metoda *add*. Naravno, ukoliko je BorderLayout manager određen u panelu tada će metoda imati dva parametra (komponenta koja se dodaje u panel i njena pozicija na panelu).

U klasi JPanel postoje četiri konstruktora:

- a) *public JPanel()*
- b) *public JPanel(boolean isDoubleBuffered)*
- c) *public JPanel(LayoutManager layout)*
- d) *public JPanel(LayoutManager layout, boolean isDoubleBuffered)*

Pokrenuti Termin7i8 u Eclipsu sa ulaznim parametrom 2c. Pogledati i prokomentarisati kod klase PanelFrame. Obratiti pažnju na gornji panel i na centralni panel i na njihove layout manager-e.



7.3.4. Komponenta JSplitPane

Klasa JSplitPane je samo-sadržeci kontejner koji omogućava horizontalnu ili vertikalnu podelu na oblasti kojima korisnik može promeniti dimenzije pomeranjem pregrade koja ih razdvaja. Da bi se koristio JSplitPane potrebno ga je kreirati sa odgovarajućom orijentacijom i komponentama koje treba podeliti.

```
public JSplitPane(int newOrientation Component component1,Component  
component2)
```

Za vertikalnu podelu koponenti koristićemo sledeći konstruktor:

```
new JSplitPane(JSplitPane.VERTICAL_SPLIT,gornjaKomponenta,  
donjaKomponenta);
```

dok za horizontalnu podelu komponenti koristićemo:

```
new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,levaKomponenta,desnaKomponenta);
```

Postavljanje inicijalne pozicije lokacije pregrade se može izvršiti sledećim metodama:

```
public void setDividerLocation(double proportionalLocation)  
public void setDividerLocation(int location)
```

Takođe je moguće i pregradu načiniti „one-touch expandable“ ili otvaranje na dodir čime se omogućava korisniku da jednim klikom otvori ili zatvori jedan deo deljenog kontejnera:

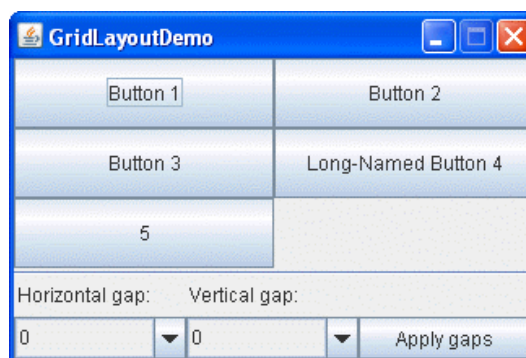
```
public void setOneTouchExpandable(boolean newValue)
```

Veoma značajno može biti i postavljanje minimalne veličine komponenti koje JSplitPane deli čime se postavlja granica do koje se pregrade može pomerati.

pokrenuti Termin7i8 u Eclipsu sa ulaznim parametrom 2d. Pogledati i prokomentarisati kod klase SplitPane.

7.3.5. GridLayout

Raspoređuje komponente u mrežu čije su ćelije jednake veličine. Svim komponentama u kontejneru se daju iste dimenzije. Ne poštuje preferirane, minimalne ni maksimalne veličine komponenti.



Za konstruktor klase GridLayout zadaje se broj vrsta i kolona. U primeru sa slike korišćeno je tri vrste i dve kolone:

```
new GridLayout (3,2);
```

pokrenuti Termin7i8 u Eclipsu sa ulaznim parametrom 4a. Pogledati i prokomentarisati kod klase GridLayoutFrame.

Pošto su ćelije grida određene dvodimenzionalnim koordinatama mreže, bilo bi za očekivati da postoji metoda *add* koja bi kao parametre imala koordinate mreže u koje bi se smeštala komponenta. Ova vrsta metode ne postoji u kontejnerima zato što se prvo u kontejner dodaju komponente a proces rapoređivanja se izvršava tek kasnije, neposredno pre nego što je GUI prikazan.

Pored navedenog konstruktoru sa 2 parametra, postoji i konstruktor sa 4 parametra, u kome se pored broja vrsta i kolona navodi i horizontalni i vertikalni razmak između komponenti. ***pokrenuti Termin7i8 u Eclipsu sa ulaznim parametrom 4b. Pogledati i prokomentarisati kod klase GridLayoutFrame.***

7.3.6. BorderLayout

Klasa `javax.swing.BoxLayout` definiše menadžer rasporeda koji uređuje komponente ili u jedan red ili u jednu kolonu. Korisnik određuje da li želi uređenje po redovima ili kolonama prilikom kreiranja objekta `BoxLayout`. Konstruktor `BoxLayout` zahteva dva parametra:

Konstruktor	Opis
<code>BoxLayout(Container, int)</code>	Container - referenca na kontejner na koji se odnosi menadžer rasporeda int - konstanta koja određuje vrstu rasporeda: BoxLayout.X_AXIS - uređenje po vrstama BoxLayout.Y_AXIS - uređenje po kolonama

Komponente se dodaju u vrstu sleva nadesno, ili u kolonu odozgo na dole. Komponente se ne prelivaju u sledeću vrstu ili kolonu kada se vrsta napuni. Kada se doda više komponenti nego što može biti prihvaćeno u dostupnom prostoru, menadžer rasporeda će smanjiti veličinu komponenti ili ih čak odseći i držaće ih sve u jednoj vrsti ili koloni. Kada se radi sa vrstom komponente, Box menadžer rasporeda će pokušati da sve komponente budu jednake visine dok će poštovati željene širine. Kada se radi sa kolonom komponente Box menadžer će pokušati da sve komponente budu iste širine poštujući željene visine.

```
JPanel panBottom=new JPanel();
BoxLayout box=new BoxLayout(panBottom, BoxLayout.X_AXIS);
panBottom.setLayout(box);
```

```
JButton btnOk=new JButton("Ok");
btnOk.setPreferredSize(new Dimension(75,25));
```

```
JButton btnCancel=new JButton("Cancel");
btnCancel.setPreferredSize(new Dimension(75,25));
```

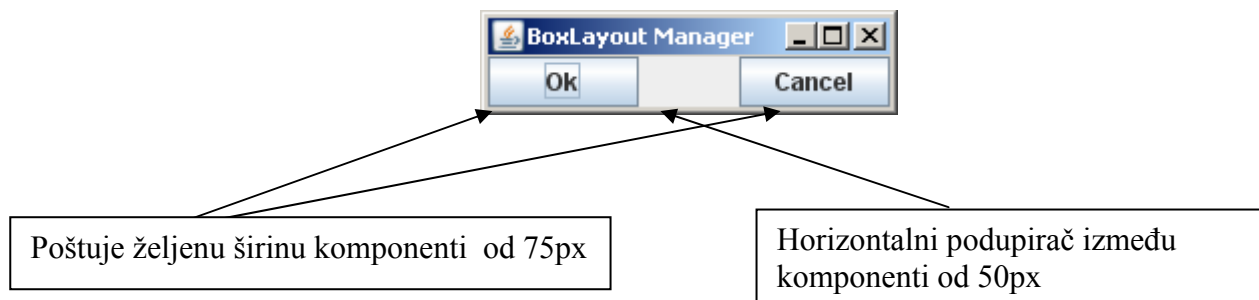
Klasa **Box** sadrži statičke metode za pravljenje nevidljive komponente koja se naziva podupirač (**strut**). Vertikalni podupirač ima datu visinu u pikselima i nultu širinu. Horizontalni podupirač ima datu širinu u pikselima i nultu visinu. Svrha ovih podupirača je da vam omoguće da ubacite prazan prostor između komponenti. Postavljanjem horizontalnog podupirača između dve komponente u horizontalno uređenom Box kontejneru fiksirate razdaljinu između komponenti.

Metoda	Opis
<code>Component createHorizontalStrut(int)</code>	Pravi horizontalni podupirač zadate širine
<code>Component createVerticalStrut(int)</code>	Pravi vertikalni podupirač zadate visine

```
JPanel panBottom=new JPanel();
BoxLayout box=new BoxLayout(panBottom, BoxLayout.X_AXIS);
panBottom.setLayout(box);
```

```
JButton btnOk=new JButton("Ok");
btnOk.setPreferredSize(new Dimension(75,25));
```

```
JButton btnCancel=new JButton("Cancel");
btnCancel.setPreferredSize(new Dimension(75,25));
panBottom.add(btnOk);
panBottom.add(Box.createHorizontalStrut(50));
panBottom.add(btnCancel);
```



Raspodela preostalog prostora u objektu Box može se kontrolisati lepkom (**glue**). Lepak je nevidljiva komponenta čija je jedina funkcija zauzimanje preostalog prostora u Box kontejneru. Lepak obezbeđuje elastičnu vezu između dve komponente koja se po potrebi može proširiti ili skupiti pa deluje kao „opruga“. Komponente lepka se mogu postaviti na jedan ili oba kraja između komponenti u Box kontejneru. Bilo koji preostali prostor između komponenti se raspoređuje između dodatih lepka komponenti. Ako na primer želimo da sav preostali prostor bude na početku Box kontejnera trebala bi se dodati jedna lepka komponenta u kontejner.

Metoda	Opis
Component createHorizontalGlue()	Metoda dodaje horizontalni lepak u kontejner
Component createVerticalGlue()	Metoda dodaje vertikalni lepak u kontejner
Component createGlue()	Metoda dodaje horizontalni i vertikalni lepak u kontejner

```
JPanel panBottom=new JPanel();
BoxLayout box=new BoxLayout(panBottom, BoxLayout.X_AXIS);
panBottom.setLayout(box);
```

```
JButton btnOk=new JButton("Ok");
btnOk.setPreferredSize(new Dimension(75,25));
```

```
JButton btnCancel=new JButton("Cancel");
btnCancel.setPreferredSize(new Dimension(75,25));
panBottom.add(Box.createGlue());
panBottom.add(btnOk);
panBottom.add(Box.createHorizontalStrut(50));
panBottom.add(btnCancel);
```

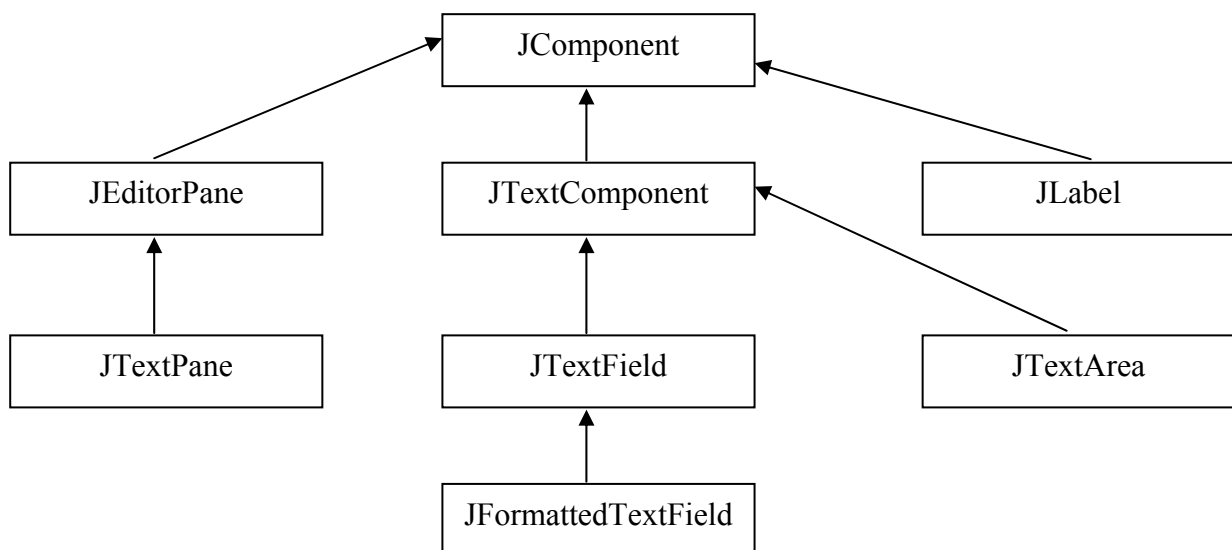


Iz primera se vidi da svaka promena veličine kontejnera koji sadrži BoxLayout prouzrokuje da sav slobodan prostor odlazi ispred prve komponente, ispred koje je dodan lepak.

pokrenuti Termin7i8 u Eclipsu sa ulaznim parametrom 6a. Pogledati i prokomentarisati kod klase BoxLayoutFrame.

7.4. Tekst komponente

Klase u paketu javax.swing koje predstavljaju tekst komponente kao i sve druge Swing komponente imaju klasu JComponent za osnovu.



Klasa **JTextArea** definiše komponentu koja dozvoljava uređivanje teksta u više linija.



Metoda	Opis
JTextArea()	Konstruktor, kreira se JTextArea objekat bez inicijalnog teksta koji sadrži
JTextArea(String)	Konstruktor, kreira se JTextArea objekat sa inicijalnim tekstom koji sadrži komponenta.
JTextArea(String, int, int)	Konstruktor, kreira se JTextArea objekat sa inicijalnim tekstom koji sadrži komponenta i zadatim brojem kolona i brojem vrsta.
void setText(String)	Postavljanje teksta koji je vidljiv u komponenti.
String getText()	Tekst koji se nalazi u komponenti.
void setEditable(boolean)	Postavljanje komponente kao dostupne/nedostupne za editovanje teksta koji se u njoj nalazi
boolean isEditable()	Vraća indikator da li je komponenta dostupna/nedostupna za editovanje teksta.
int setLineWrap(boolean)	Postavljanje prelamanja linije teksta u komponenti ukoliko je suviše dugačka. Inicijalno prelamanje nije uključeno pa ukoliko je linija teksta suviše dugačka tekst se jednostavno neće videti.

Komponenta JTextArea ne podržava pomeranje (skrolovanje) sadržaja, ali se dodavanje traka za pomeranje sadržaja lako realizuje - smeštanjem JTextArea komponente u JScrollPane kontejner.

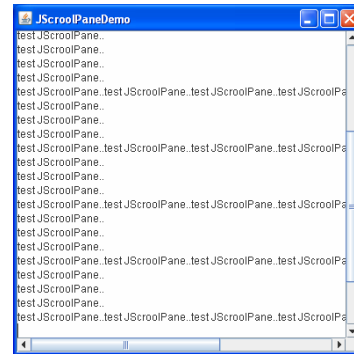
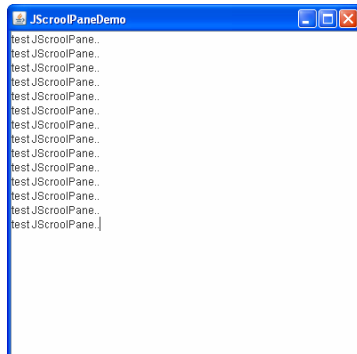
Pokrenuti Termin7i8 u Eclipsu sa ulaznim parametrom 7. Pogledati i prokomentarisati kod klase TextAreaFrame.

U slučaju da je potrebno urediti samo jednu liniju teksta koristi se klasa **JTextField**. Ovu komponentu je potrebno samostalno proučiti (<http://download.oracle.com/javase/tutorial/uiswing/components/textfield.html>).

Komponente **JEditorPane** i **JTextPane** su drugačijeg stepena složenosti u odnosu na ostale tekstalne komponente : JEditorPane podržava uređivanje običnog teksta, teksta u formatima HTML i RTF(Rich Text Format). JTextPane proširuje JEditorPane i dozvoljava da se unutar teksta kojim komponenta upravlja ugrade slike i ostale komponente. Sa ovim komponentama nećemo raditi.

7.5. Komponenta JScrollPane

Ova komponenta se ponaša kao kontejner za jednu komponentu, obezbeđujući podršku za skrolovanje za tu komponentu ukoliko ona bude veća od prostora koji joj je dostupan u JScrollPane.



Komponenta JScrollPane ima 3 konstruktora:

Metoda	Opis
JScrollPane()	Konstruktor, kreira komponentu JScrollPane bez komponente koja se može pomerati korišćenjem klizača
JScrollPane(Component view)	Konstruktor, kreira komponentu JScrollPane, sa komponentom koja se može pomerati korišćenjem klizača
JScrollPane(Component view, int vsbPolicy, int hsbPolicy)	Konstruktor, kreira komponentu JScrollPane, sa komponentom koja se može pomerati korišćenjem klizača uz postavljanje predifisanog ponašanja vertikalnog i horizontalnog klizača.

Podrazumeva se da JScrollPane ne prikazuje klizače sve dok to nije neophodno. Ovo ponašanje je moguće promeniti sa korišćenjem 3. konstruktora i prosleđivanja sledećih konstanti:

Moguće vrednosti parametra vsbPolicy su sledeće konstatnte:

- * ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED
- * ScrollPaneConstants.VERTICAL_SCROLLBAR_NEVER
- * ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS

Moguće vrednosti parametra hsbPolicy su sledeće konstatnte:

- * ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED
- * ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER
- * ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS

Ukoliko se koristi konstruktor se jednim parametrom tada se vertikalni i horizontalni skroleri pojavljuju po potrebi (
vsbPolicy= VERTICAL_SCROLLBAR_AS_NEEDED i
hsbPolicy= HORIZONTAL_SCROLLBAR_AS_NEEDED)

pokrenuti Termin7i8 u Eclipsu sa ulaznim parametrom 9. Pogledati i prokomentarisati kod klase ScroolPaneDemo.

7.6. Zadaci

Realizovati aplikaciju kao na slici. Obratiti pažnju da se između leve JTextArea komponente i desnog panela nalazi JScrollPane komponenta.

