

5.1. Komponente i kontejneri u Swing biblioteci

5.2. „Top level“ kontejneri i komponentalna hijerarhija

5.3. Osnove Swing komponenti

5.4. Paneli

5.5. Dugmad

5.6. Labele

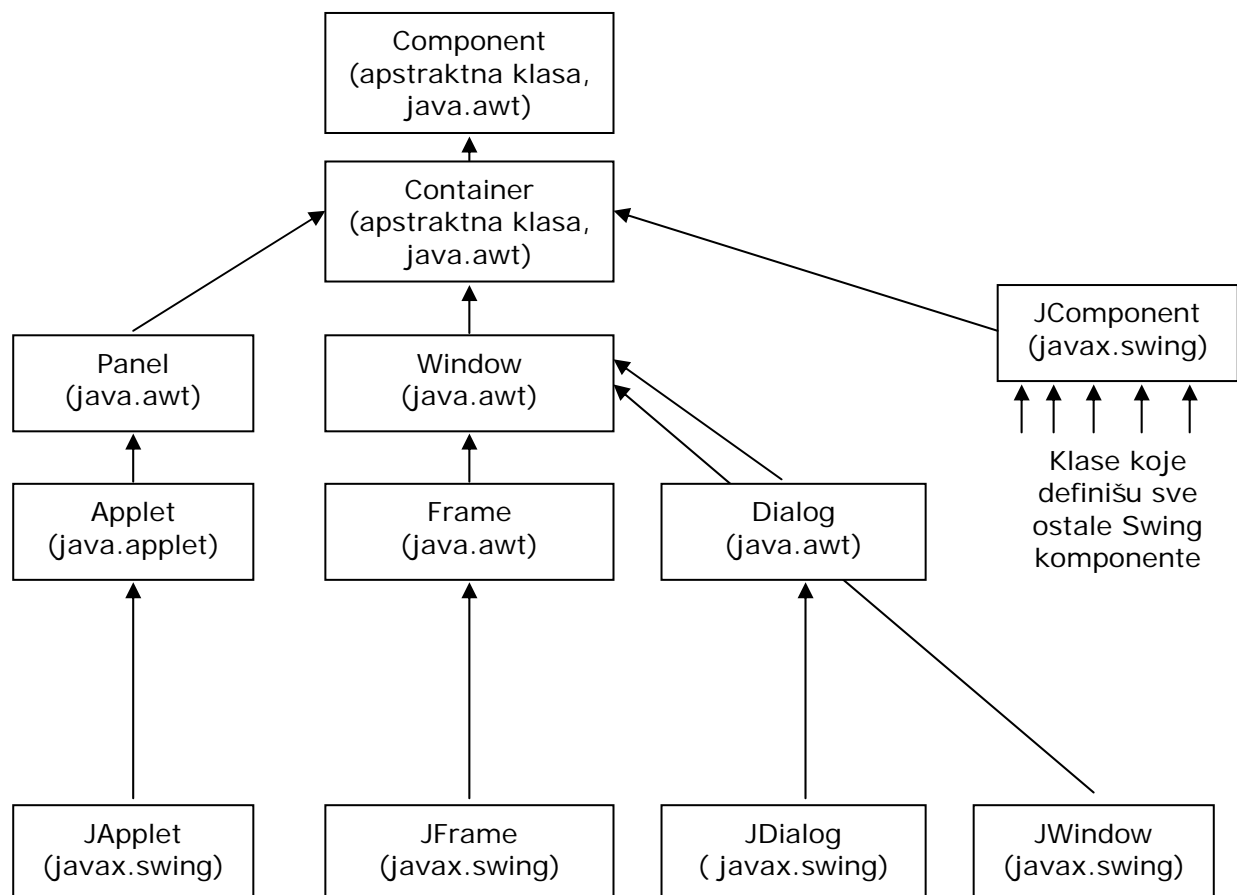
5.7. Meniji

5.8. Toolbar

5.9. Zadaci

5.1. Komponente i kontejneri u Swing biblioteci

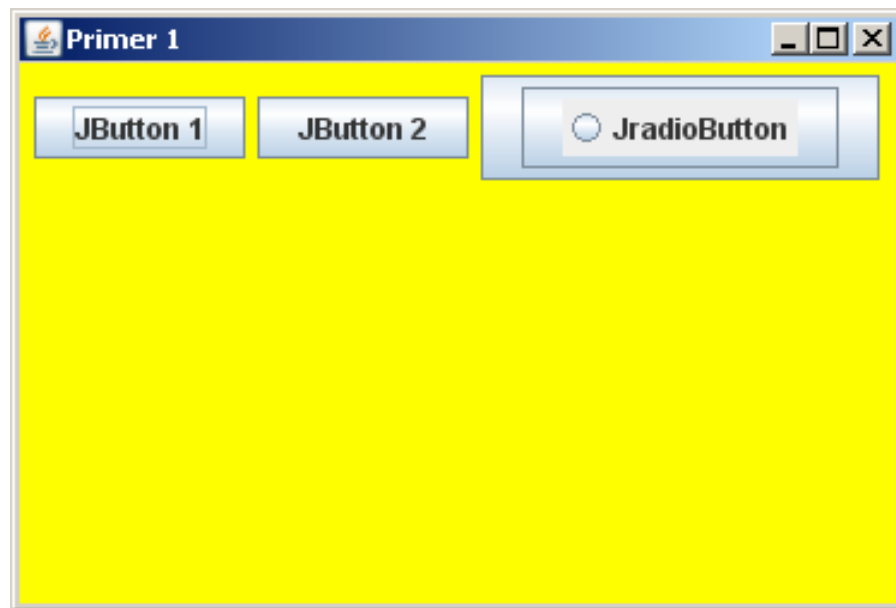
Uopšteno govoreći komponenta predstavlja grafički entitet određene vrste koji se može prikazati na ekranu (dugme, meni, toolbar...). U programskom jeziku Java, komponenta je objekat (instancija) bilo koje klase koja je potklasa (naslednik) klase `Component`. Jednu vrstu komponente smo već upoznali - to je klasa `JFrame`. Pregled potrebnih komponenti započecemo prikazom dela hijerarhije između klasa koje predstavljaju komponente.



Sve klase izvedene iz klase `java.awt.Container` mogu da sadrže druge objekte bilo koje od klasa izvedenih iz klase `java.awt.Component` i uopšteno se nazivaju **kontejneri**. Kako je klasa `java.awt.Container` potklasa klase `java.awt.Component`, svaki kontejner objekat je istovremeno i `Component` objekat, pa svaki kontejner može sadržati druge kontejnere. **Izuzetak je klasa `java.awt.Window` i njene potklase koji ne mogu biti sadržani u drugom kontejneru.** Klasa `javax.swing.JComponent` je osnov svih Swing komponenti, pa kako je ova klasa izvedena iz klase `Container`, sve Swing komponente su takođe kontejneri.

Šta znači to da jedna klasa može da sadrži objekte druge klase, odnosno šta to čini specifičnim naslednike klase `java.awt.Container` u koje spadaju sve Swing komponente?

Primer 1. Na materijalima nalazi se projekat Termin5i6. Izvršiti import ovog projekta u Eclipse. U projektu se nalazi nekoliko različitih primera za današnji termin i u zavisnosti od unete vrednosti ulaznog argumenta biće prikazan JFrame sa odgovarajućim primerom (argumenti se u Eclipse-u unese u delu Run -> Run configurations->Arguments->Program arguments). Za primer 1 uneti vrednost ulaznog argumenta „1“ i pokrenuti aplikaciju. ***Pogledati i prokomentarisati kod klase MyApp i Frame1.*** Kao rezultat videćemo instancu klase JFrame koja u sebi sadrži žuti panel (objašnjenje u sledećem poglavlju) unutar koga se nalaze 3 dugmeta. Treće dugme u sebi sadrži 4. dugme koje u sebi sadrži radio dugme.



Kontejner je bilo koja komponenta koja ima u osnovi klasu `java.awt.Container`, tako da su sve Swing komponente kontejneri. Klasa `Container` obezbeđuje mogućnost da sadrži druge komponente. Kako je klasa `Container` klasa tip `abstract`, nije moguće napraviti instancu klase `Container`. Umesto toga objekti potklase, kao što su `JFrame` ili `JDialog`, nasleđuju sposobnost da sadrže druge komponente. Komponente u kontejneru prikazane su unutar oblasti koju na ekranu zauzima kontejner. Upravljanje rasporedom komponenti unutar kontejnera zavisi od **LayoutManager**-a koje ćemo detaljno razmotriti na sledećem terminu.

Komponente smeštene u kontejner zapisuju se u niz unutar objekta `Container`. Kada je potrebno, niz povećava veličinu da bi primio onoliko komponenti koliko je prisutno. Dodavanje komponente u kontejner izvršava se korišćenjem metode **`add(Component c)`**. Dodavanje komponente ne uklanja nijednu komponentu koja je već u kontejneru. Jedna komponenta može biti samo u jednom kontejneru u bilo kom trenutku. Dodavanje komponente koja je već u jednom kontejneru u drugi kontejner ukloniće je iz onog prvog kontejnera.

5.2. „Top level“ kontejneri i komponentalna hijerarhija

Svaki program koji koristi Swing komponente sadrži najmanje jedan top level kontejner. Top level kontejner predstavlja koren komponentalne hijerarhije, odnosno hijerarhije svih komponenti koje se pojavljuju unutar top level komponente. Po pravilu aplikacije koje sadrže Swing GUI komponente imaju najmanje jednu komponentalnu hijerarhiju čiji je koren instanca klase JFrame. Ova instanca predstavlja glavni prozor aplikacije.

Swing biblioteka sadrži 5 top-level kontejnerskih klasa :

Klasa	Upotreba
JFrame	Koristi se kao osnovni Java aplikacioni prozor. Objekat ove klase ima naslovnu liniju (title) i osnove za dodavanje menija.
JDialog	Ovu klasu ćemo koristiti za definisanje prozora dijaloga koji se koristi za unos podataka u program. Kreiranje dijaloga se najčešće javlja kao odgovor na neku stavku u meniju ili neko dugme koje je izabrano. Pošto je klasa JDialog direktno izvedena iz klase Window, objekat JDialog se može stvoriti u aplikaciji samo u kontekstu objekta JFrame.
JApplet	Osnovna klasa za Java 2 aplet - program koji je dizajniran da se izvršava ugrađen u web browser-ima. Primer apleta: http://richardbowles.tripod.com/java/calc.htm Nećemo koristiti ovu klasu.
JWindow	Objekat ove klase je prozor bez naslovne linije i ikonice za upravljanje prozorom. Nećemo direktno koristiti ovu klasu.
JInternalFrame	Klasa za kreiranje internih prozora u MDI (Multiple Document Interface) aplikacijama. Nećemo koristiti ovu klasu.

Zajedničke osobine top-level kontejnerskih klasa:

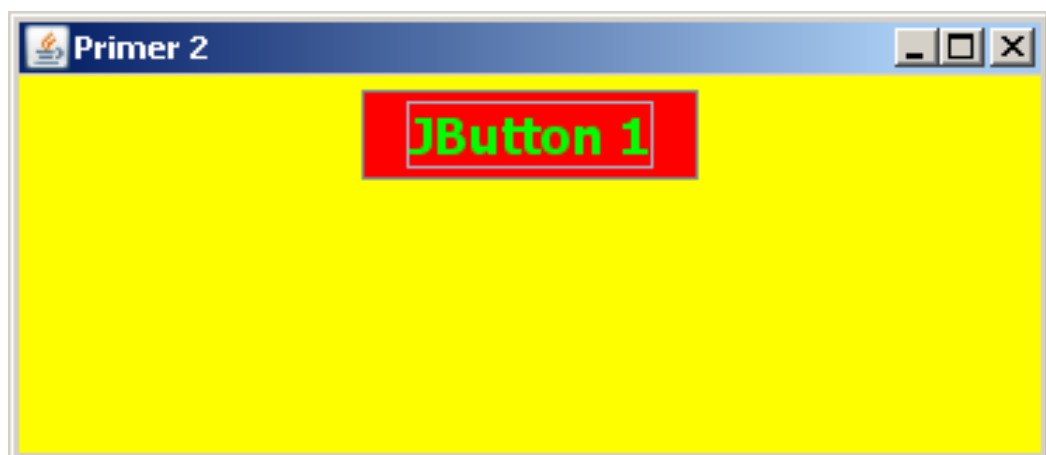
1. Da bi se pojavila na ekranu svaka GUI komponenta mora biti sadržana u komponentalnoj hijerarhiji. Komponentalna hijerarhija je stablo komponenti koje kao početni čvor ima jednu od top-level kontejnerskih klasa.
2. Svaka GUI komponenta može biti sadržana samo jednom. Ako se komponenta već nalazi u kontejneru i pokuša se dodati u neki drugi kontejner, komponenta će biti uklonjena iz prvog kontejnera i dodata u drugi kontejner.
3. Svaki top-level kontejner poseduje „content pane“ koji sadrži direktno ili indirektno sve vidljive GUI komponente koje nalaze na top-level kontejneru (panele top - level kontejnera nećemo detaljno razmatrati: <http://download.oracle.com/javase/tutorial/uiswing/components/rootpane.html>)
4. Izuzetak u smeštanju GUI komponenti u content pane predstavlja komponenta menu bar koja se smešta direktno na top level kontejner izvan content pane-a.

5.3. Osnove Swing komponenti

Kao što je već rečeno **komponentom** u Javi smatra se svaka klase koja je naslednik klase `java.awt.Component`. U zavisnosti od vrednosti atributa unutar klase `java.awt.Component` vršiće se vizuelni prikaz komponente unutar kontejnera u kome se nalazi. Klasa `Component` definiše attribute koji zapisuju sledeće informacije o objektu (ovo je zajedničko za sve komponente u Javi jer sve nasleđuju ovu klasu):

Pozicija	Position	Pozicija komponente se pamti kao (x,y) koordinata. Pozicija fiksira gde se objekat nalazi u odnosu na svoj kontejner u koordinatnom sistemu kontejnera.
Ime	Name	Naziv komponente se čuva u objektu <code>String</code> .
Veličina	Size	Veličinu komponente određuju celobrojne vrednosti za širinu i visinu objekta.
Prednja boja	Foreground	Boja prednje strane. Ova boja se koristi za bilo šta što se pojavljuje na osnovnoj komponenti, kao što je naziv na dugmetu.
Boja pozadine	Background	Boja pozadine sa kojom će objekat biti prikazan. Boja pozadine se koristi za osnovnu komponentu.
Font	Font	U slučaju da se unutar komponente nalazi tekst određuju <code>Font</code> kojim se prikazuju tekst.
Kursor	Cursor	Definiše izgled kursora kada se nađe iznad objekta.
Dostupnost	Enabled	Ako je komponenta dostupna ima normalan prikaz dok u slučaju da nije dostupna biće prikazana sivom bojom
Vidljivost	Visible	Ako je objekat označen da nije vidljiv neće biti prikazan na ekranu.

Primer 2. Pokrenuti projekat Termin5i6 sa vrednošću argumenta 2. U primeru se vidi jedna instanca klase `JButton` (naslednik klase `java.awt.Component`) sa promenjenim vrednostima osnovnih atributa. ***Pogledati i prokomentarisati kod klase `Frame2`.***



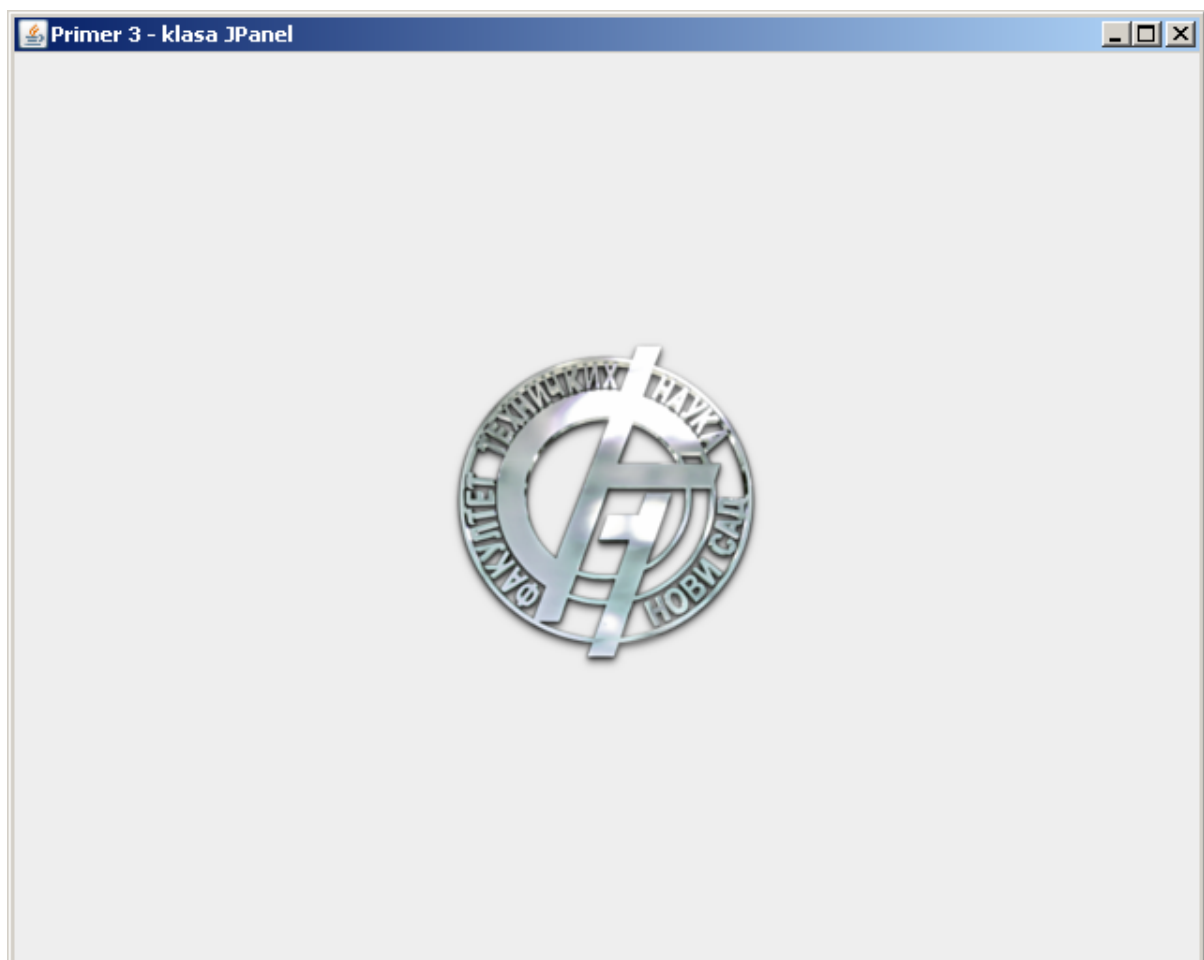
Kako su svi atributi klase Component privatni, vrednosti atributa je moguće menjati samo pozivom njegovih metoda.

Dobijanje i promena imena komponente (ime komponente ima samo opisnu vrednost i retko se javlja potreba za ovim metodama)	
<code>void setName(String name)</code>	Naziv komponente menja u prosleđeni name
<code>String getName()</code>	Vraća naziv komponente
Dobijanje i promena vidljivosti komponente	
<code>void setVisible(boolean visible)</code>	Određuje vidljivost komponente na ekranu. Komponenta bez obzira na vidljivost ostaje u kontejneru u kome se nalazi
<code>boolean isVisible()</code>	Vraća trenutnu vidljivost komponente.
Dobijanje i promena dostupnosti komponente	
<code>void setEnabled(boolean visible)</code>	Menja dostupnost komponente.
<code>boolean isEnabled()</code>	Vraća trenutnu dostupnost komponente
Veličina i pozicija komponente (iako postoji veći broj predefinisanih metoda, promena veličine i pozicije komponente u kontejneru se nikad neće raditi sa ovim metodama već korišćenjem LayoutManager-a)	
<code>Dimension getSize()</code>	Vraća dimenzije komponente u obliku objekta tipa Dimension
<code>Point getLocation()</code>	Vraća poziciju komponente u obliku objekta tipa Point
<code>Rectangle getBounds()</code>	Vraća poziciju i dimenzije komponente u obliku objekta tipa Dimension
<code>void setSize(Dimension d)</code>	Postavlja veličinu komponente u dimezije prosleđenog objekta Dimension
<code>void setLocation(int x, int y)</code>	Postavlja poziciju na lokaciju prosleđenih x i y koordinata
<code>void setBounds(int x, int y, int width, int height)</code>	Postavlja veličinu i poziciju komponente na prosleđene koordinate i dimenzije
Vizuelne karakteristike komponente	
<code>void setBackground(Color color)</code>	Pogledati klasu java.awt.Color
<code>Color getBackground()</code>	Pogledati klasu java.awt.Color
<code>void setForeground(Color color)</code>	Pogledati klasu java.awt.Color
<code>Color getForeground()</code>	Pogledati klasu java.awt.Color
<code>void setCursor(Cursor cursor)</code>	Pogledati klasu java.awt.Cursor
<code>Cursor getCursor()</code>	Pogledati klasu java.awt.Cursor
<code>void setFont(Font font)</code>	Pogledati klasu java.awt.Font
<code>Font getFont()</code>	Pogledati klasu java.awt.Font

5.4. Paneli

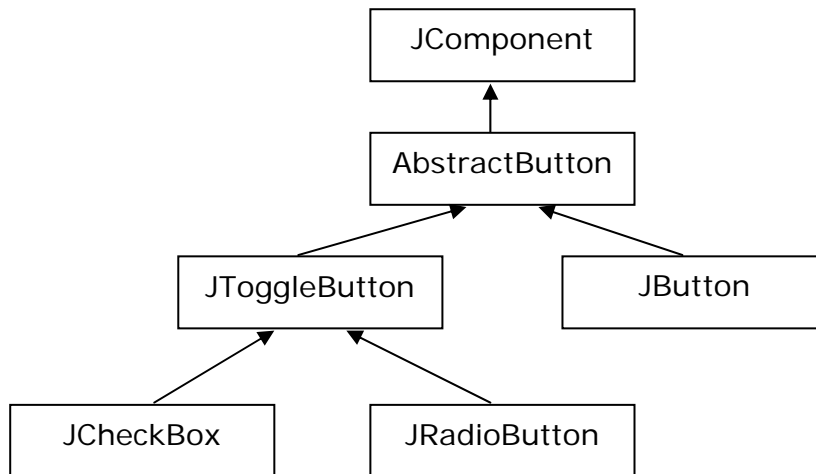
Klasa JPanel nudi klasu kontejnera niže kategorije u odnosu na „Top level“ kontejnere. Da bi se grupisale komponente u celinu kada se uređuje ekran može koristiti i klasa JPanel. Prednost ove klase u odnosu na top-level kontejnere je to što komponente u JPanelu mogu biti iscrtane na duplo baferisani način (duplo baferisanje osigurava da korisnici ne vide delimično osvežavanje ekrana između scena) i možemo postaviti različite vrste ivica (border-a) oko panela. Ipak najznačajnija prednost klase JPanel je ta što je moguće veći broj komponenti smeštati u panele koji se zatim grupišu u okviru top - level kontejnera. Ovo će biti razmatrano u sledećem terminu prilikom rada sa LayoutManager-ima.

Primer 3. Pokrenuti projekat Termin5i6 sa vrednošću argumenta 3. U primeru je prikazan rad sa klasom JPanel u kome je integrisana slika koja se prikazuje kao pozadina panela koji je smeštan u JFrame. Ovaj koncept će biti iskorišćen prilikom realizacije prvog projekta kao logo aplikacije. ***Pogledati i prokomentarisati kod klase Frame3.***



5.5. Dugmad

Swing klase dugmadi u paketu javax.swing definišu razne vrste dugmadi kojima se operiše pritiskom miša. Klase dugmadi u osnovi imaju klasu AbstractButton:



JButton	Obično dugme za pritiskanje.
JToggleButton	Definiše dugme sa dva stanja, pritisnuto ili ne i postoje još dve specijalizacije ove klase (JCheckBox i JRadioButton)
JCheckBox	Dugme sa kvadratnim poljem za potvrđivanje. Pritiskom mišem na poje za potvrđivanje menja se njegovo stanje
JRadioButton	Radio dugmad obično rade u grupi, tako da samo jedno od njih može biti u pritisnutom stanju u bilo kom trenutku. Udruživanje se uspostavlja dodavanjem objekta JRadioButton u objekat ButtonGroup koji vodi računa o stanju dugmadi u grupi

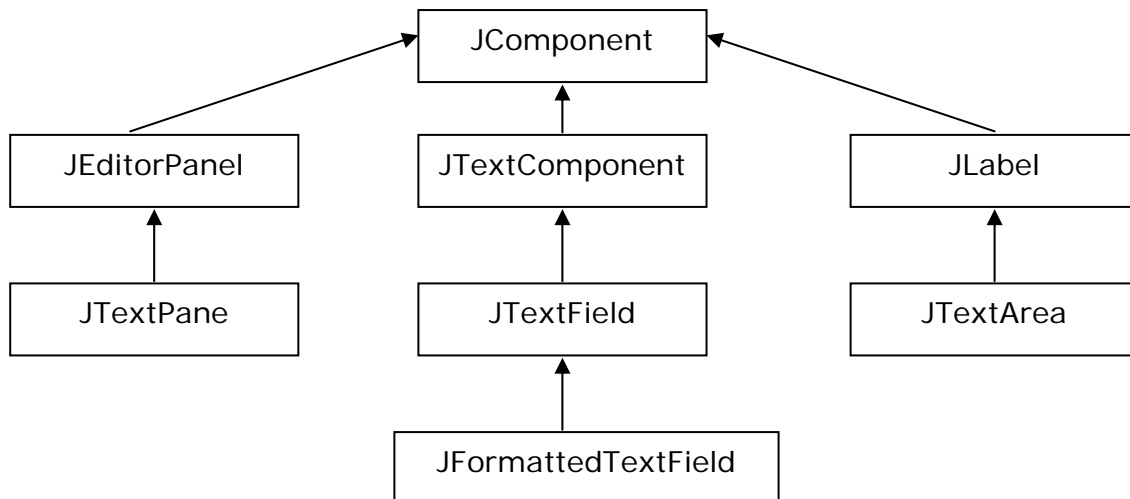
Sva dugmad mogu da se prikažu sa tekstualnim nazivom, ikonom ili oba.

Primer 4. Pokrenuti projekat Termin5i6 sa vrednosti ulaznog argumenta 4. ***Pogledati i prokomentarisati kod klase Frame4***



5.6. Labele

Labele spadaju u tekst komponente koji imaju velike mogućnosti u okviru Swing komponenti. Klase u paketu javax.swing koje predstavljaju tekst komponente prikazane su na slici i one kao i ostale Swing komponente imaju klasu JComponent kao osnovu.



Najosnovnija tekst komponenta je objekat JLabel. Komponenta tipa JLabel je pasivna i ne reaguje na ulazne događaje, tj. tekst u njoj se ne može menjati. Ostale tekst komponente će biti objašnjenje po potrebi.

Primer 5. Pokrenuti projekat Termin5i6 sa vrednosti ulaznog argumenta 5. Obratiti pažnju na metodu `setToolTipText(String text)` koja se nalazi u klasi JComponent i koja se može koristiti u svim Swing komponentama. ***Pogledati i prokomentarisati kod klase Frame5***

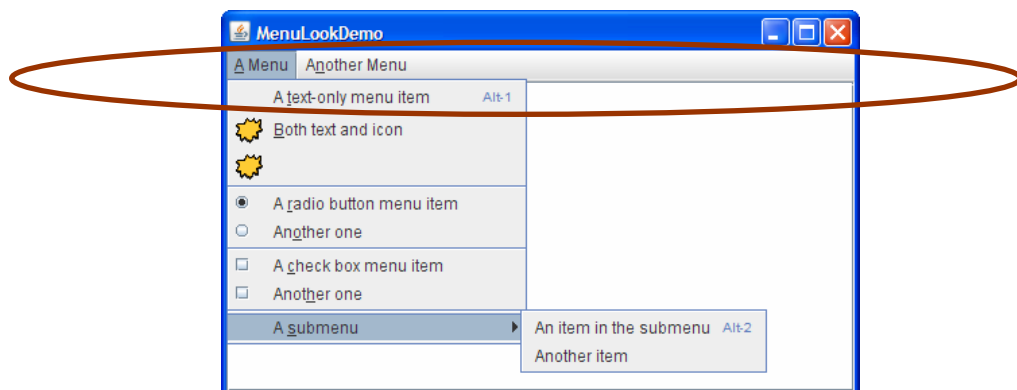


5.6. Meniji

Meni predstavlja seriju padajućih lista koji se koriste za izvršavanje odgovarajućih zadataka. Meniji često dupliraju isti zadatak koji se može izvršiti korišćenjem tastera ili nekih drugih komponenata korisničkog interfejsa, omogućavajući korisniku programa da na više načina obavi isti zadatak (obično se uz glavni meni u aplikaciji pojavljuje i toolbar kao i tastaturne prečice).

Meniji se u programskom jeziku Java implementiraju korišćenjem sledeće tri komponente:

- 1) **JMenuItem** - stavka menija, ima ulogu sličnu kao i obično dugme (JButton) , odnosno klikom na JMenuItem očekuje se izvršavanje neke akcije. Može da prikazuje tekst i/ili ikonu. Postoje dve specijalizacije (naslednika) ove klase: **JRadioButtonMenuItem** i **JCheckBoxMenuItem** koji imaju izgled i ulogu analognu komponentama JRadioButton i JCheckBox.
- 2) **JMenu** - padajući meni koji sadrži jednu ili više JMenuItem komponenti ili njenih naslednika (JRadioButtonMenuItem ili JCheckBoxMenuItem) , separatora (linija koja razdvaja pojedine stavke menija) kao i drugih JMenu komponenti koje predstavljaju podmeni.
- 3) **JMenuBar** - kontejner koji sadrži jednu ili više JMenu komponenti i koji prikazuje njihove nazive. Obično se nalazi ispod linije naslova u glavnoj formi aplikacije.



Na slici je zaokružena instanca klase **JMenuBar** koja sadrži dve instance klase **JMenu** (A Menu i Another Menu).

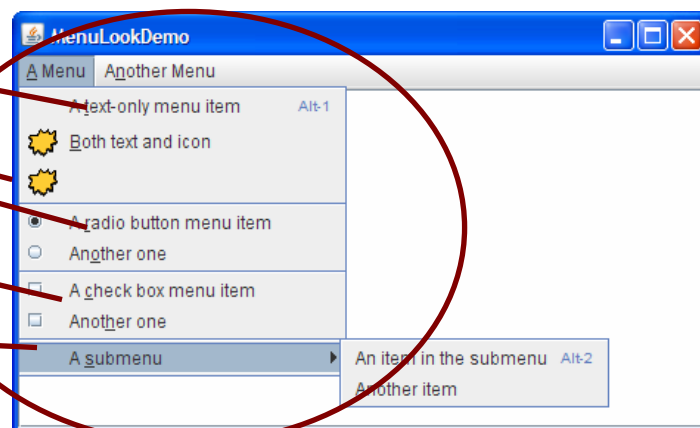
JMenuItem

Separator

JRadioButtonMenuItem

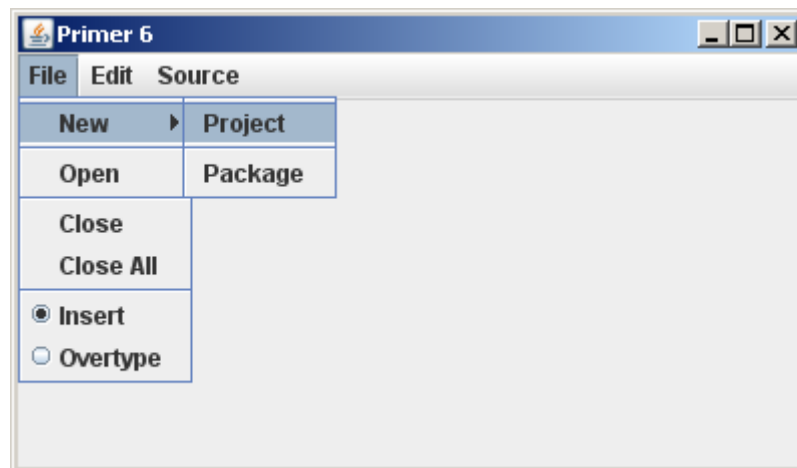
JCheckBoxMenuItem

JMenu



Na slici je zaokružena instanca klase **JMenu** (A menu) koja predstavlja padajući meni i sadrži instance klase **JMenuItem** (prve tri stavke), instance klase **JRadioButtonMenuItem** i **JCheckBoxMenuItem**, instancu klase **JMenu** koja predstavlja podmeni. Pojedine stavke menija su razdvojene **separatorom**.

Primer 6. Pokrenuti projekat Termin5i6 sa argumentom „6“. Aplikacija će prikazati JFrame koji u sebi sadrži jednostavan glavni meni aplikacije. (**Pogledati i prokomentarisati kod klase Frame6 i MyMenuBar**).



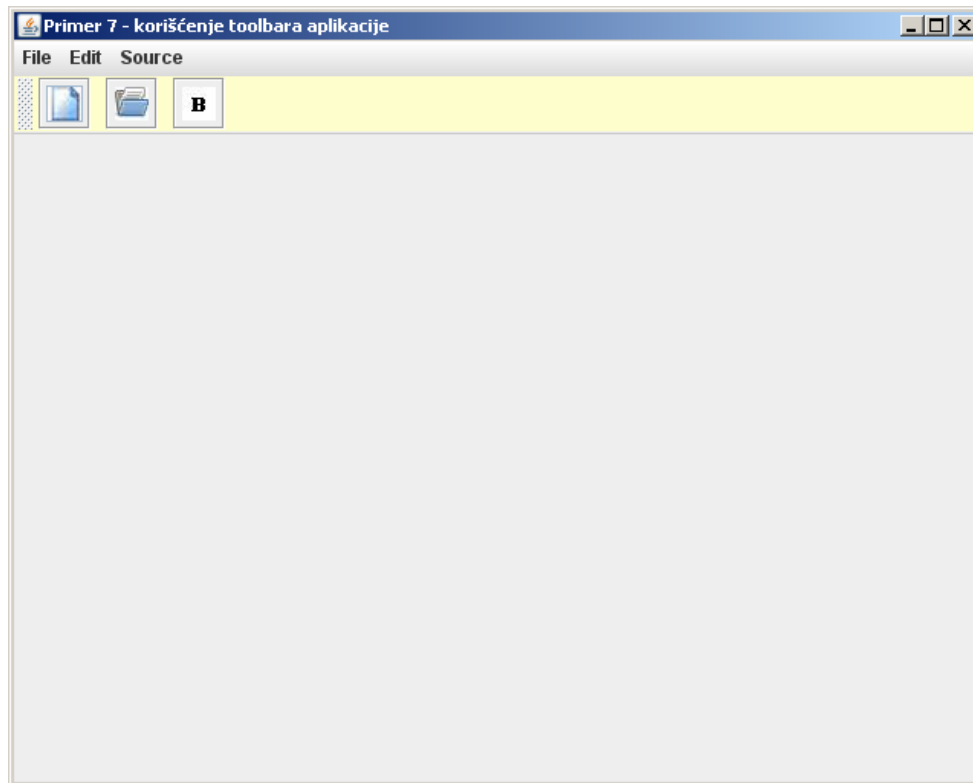
Dodavanje menija (instance klase JMenuBar) u glavni prozor aplikacije se izvršava pozivom metode setJMenuBar(JMenuBar) klase JFrame. Ova metoda postoji u klasama JFrame i JDialog i obezbeđuje postojanje samo jednog glavnog menija u aktivnom prozoru aplikacije.

5.7. Toolbar

Toolbar (paleta sa alatkama) kreira se u Javi korišćenjem javax.swing.JToolBar klase. Toolbar predstavlja kontejner koji grupiše nekoliko komponenti u jednu vrstu ili kolonu. Toolbar se sastoji od redova ili kolona komponenti koje predstavljaju grupu naredbi aplikacije koje se najčešće upotrebljavaju. Podrazumeva se da su palete sa alatkama horizontalne ali se orijentacija (horizontalno ili vertikalno) može eksplicitno podešavati. Nakon što se kreira toolbar neophodno je u njega dodati željene komponente korišćenjem metode **add(Component c)**, gde c predstavlja komponentu koja treba da se nađe u paleti sa alatkama. Mnogi programi koji sadrže toolbar omogućavaju korisnicima da ih pomeraju. Tada je reč o pokretnim toolbar-ovima pošto se mogu postavljati uz ivicu ekrana kao i u novi prozor različit od originalnog.

Za kreiranje toolbar-a u Swingu koristi se klasa javax.swing.JToolBar.JToolBar podržava smeštanje na bilo koju stranu prozora. Moguć unos bilo koje komponente na toolbar, podrazumeva se JButton. Dugmići na Toolbar-u mogu imati sliku, natpis ili oboje.

Primer 7. Pokrenuti projekat Termin5i6 sa argumentom „7“. Aplikacija će prikazati JFrame koji u sebi sadrži jednostavan glavni meni aplikacije i pokretljivi toolbar. (*Pogledati i prokomentarisati kod klasa **Frame7** i **MyToolbar***).

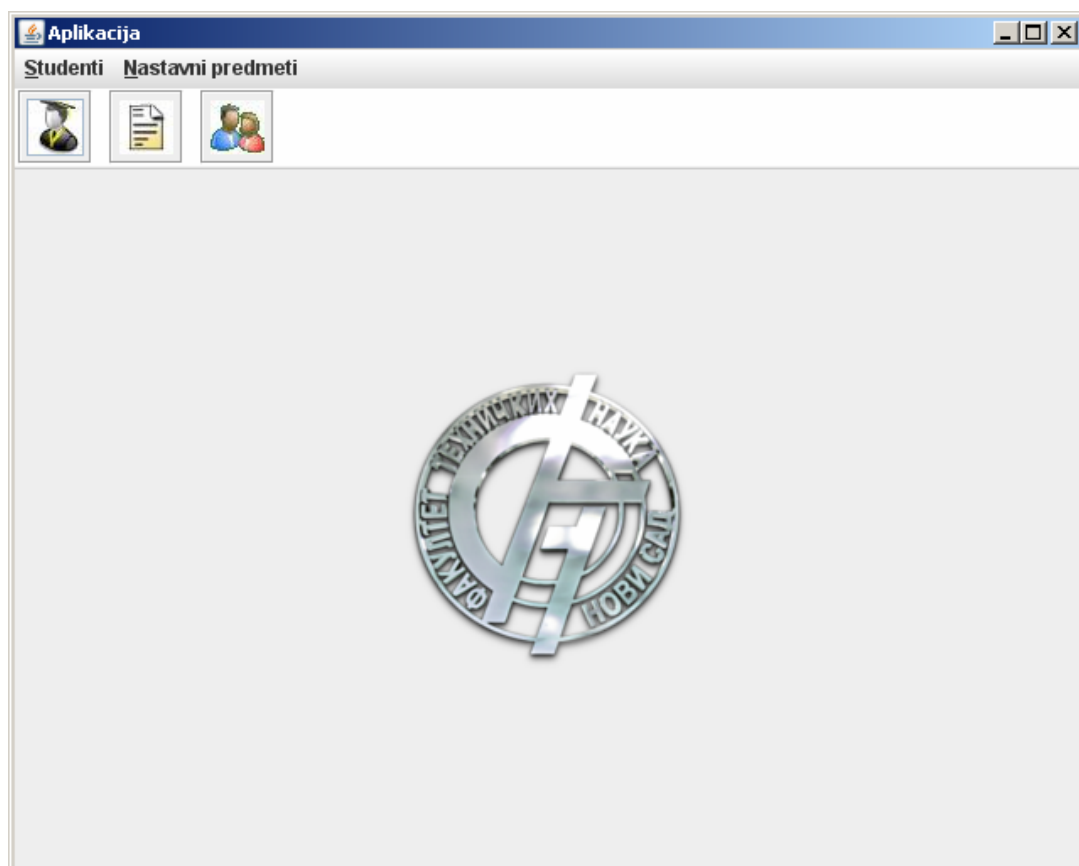


5.8. Zadatak

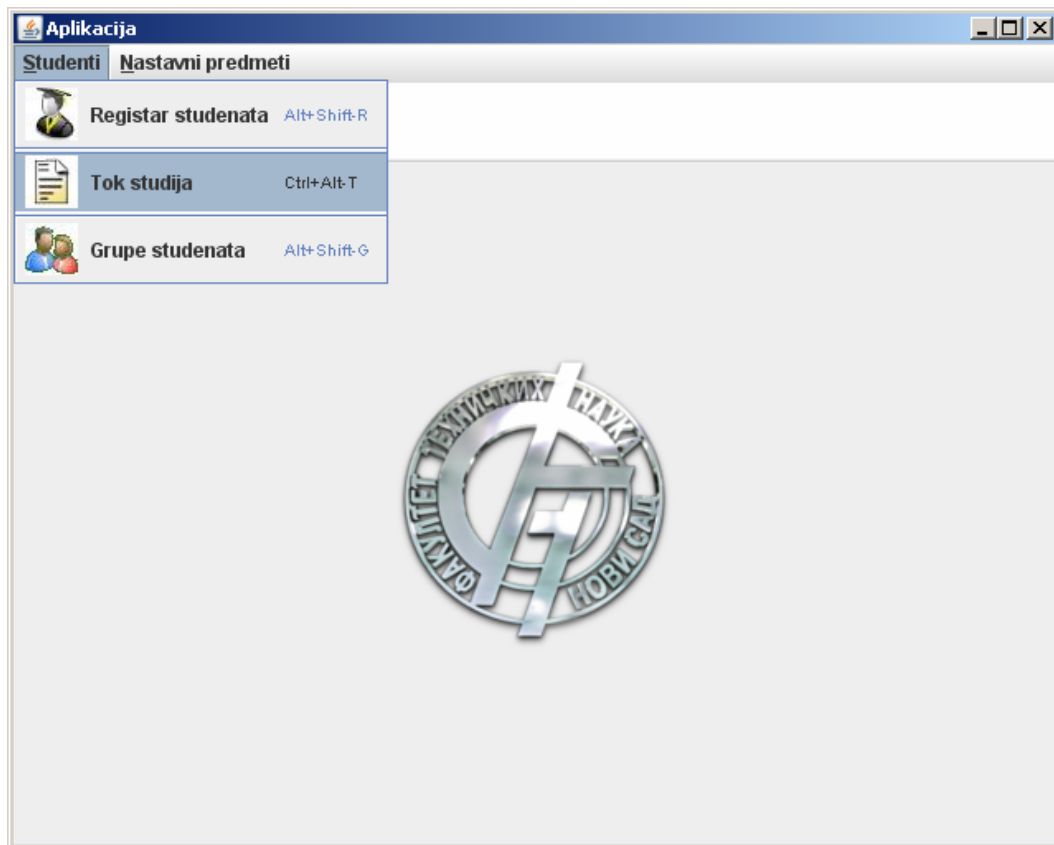
Meniji kako u Javi tako i u ostalim programskim jezicima sa razvijenom podrškom za GUI pružaju mogućnost alternativnog korišćenja menija aplikacije putem tastature:

- a) **Mnemonici** nude mogućnost navigacije kroz strukturu menija korišćenjem tastature (pogledajte u Eclipsu po jedno podvučeno slovo na svakoj stavci glavnog menija i sa npr. ALT+F aktiviraće se File meni).
- b) **Akceleratori** sa druge strane nude mogućnost zaobilaznje navigacije kroz meni i direktan pristup stavci menija (pogledajte u Eclipsu stavku File->New, ona se može aktivirati pritiskom ALT+SHIFT+N).

Kao zadatak treba proučiti način dodavanja akceleratora i mnemonika u meniju (<http://download.oracle.com/javase/tutorial/uiswing/components/menu.html#mnemonic>) i napraviti aplikaciju kao sa slike:



Obratiti pažnju na mnemonike menija



Obratiti pažnju na akceleratora menija