

3.1. JFC i Java SWING

3.2. Glavni prozor aplikacije - klasa JFrame

3.3. Klasa java.awt.Toolkit

3.4. Povezivanje JFC-a sa Java izvornim kodom

3.5. Primer složene GUI aplikacije

3.6. Zadaci

3.1. JFC i Java SWING

JFC predstavlja skraćeni naziv za osnovne java klase (Java Foundation Classes) koje između ostalog obuhvataju i grupu Java biblioteka i Java klase namenjenih za izradu grafičkog korisničkog interfejsa (GUIs - Graphical User Interfaces), dodavanje bogate grafičke funkcionalnosti kao i implementaciju interaktivnih Java aplikacija.

Osnovne karakteristike JFC-a u domenu izrade GUI -a su sledeće:

SWING GUI komponente	Obuhvataju sve standardne GUI komponente, od dugmadi do formi, dijaloga i tabela.
Podrška proširivom izgledu GUI komponenti (Pluggable Look-and-Feel Support)	Izgled Java SWING aplikacija je proširiv, dozvoljava izbor izgleda komponenti od strane korisnika biblioteka.
Java 2D API	Omogućava integraciju 2D grafike, teksta i slika u GUI aplikacije.
Internacionalizacija (Lokalizacija)	Omogućava implementaciju aplikacija koje imaju mogućnost interakcije sa korisnikom aplikacija na njihovom jeziku i sa kulturološkim specifičnostima u zavisnosti od regiona u kome se koriste.

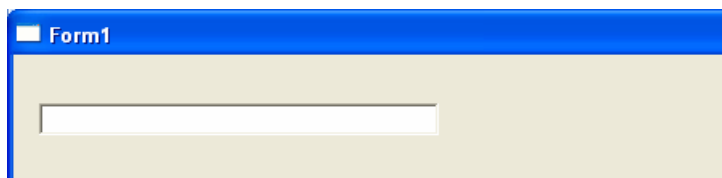
Pre svega će biti interesantan segment JFC-a koji se odnosi na rad sa SWING komponentama. SWING API sadrži ukupno 18 javnih paketa, od kojih su 2 posebno značajna:

javax.accessibility	javax.swing.border	javax.swing.event
javax.swing.plaf	javax.swing.plaf.metal	javax.swing.plaf.synth
javax.swing.text	javax.swing.text.html.parser	javax.swing.tree
javax.swing	javax.swing.colorchooser	javax.swing.filechooser
javax.swing.plaf.basic	javax.swing.plaf.multi	javax.swing.table
javax.swing.text.html	javax.swing.text.rtf	javax.swing.undo

Java sadrži dva seta GUI (Graphical User Interface) komponenti. Prvi set komponenti sadrži paket java.awt.* i koristio se u verzijama Jave 1.0 i 1.1. Drugi set komponenti predstavlja paket javax.swing.* i uveden je sa Javom 1.2. Swing zamenjuju sve awt komponente a obezbeđuje i dodatne, kompleksnije komponente. U aplikaciji je moguće istovremeno upotrebljavati i Swing i AWT komponente ali ovo nije preporučljivo jer se u određenim slučajevima neke od komponenti neće prikazati na ispravan način.

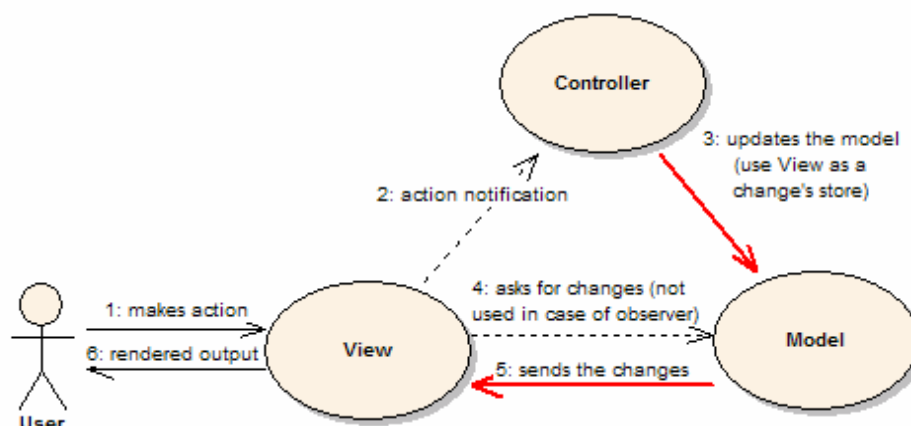
Swing je nezavisan od operativnog sistema pod kojim radi aplikacija. Znači podjednako dobro izgleda i pod WINDOWS, LINUX, MAC OS. Omogućava pluggabil-an rad, tj. korisnik može da uključi sopstvenu implementaciju određenih komponenti.

Swing podržava „**look-and- feel**“ princip. „Look-and -feel“ izraz u oblasti Dizajna softvera se odnosi na Korisnički Grafički Interfejs (GUI - Graphical User Interface) i označava odnos njegovog dizajna (boje, senke, raspored, izgled - look) i ponašanja, odnosno uloge (dugme, box, meniji - feel). „Look-and-feel“ u korisničkom interfejsu operativnih sistema na primer ima dvojaku ulogu. Prva uloga je stvaranje „brenda“ odnosno prepoznatljivog izgleda pojedinih operativnih sistema. Druga uloga je povećanje lakoće korišćenja korisničkog interfejsa, jer iskustvo sa GUI-om jednog OS omogućava rad sa drugim OS ukoliko postoji isti „look-and- feel“. Dakle cilj je da izgled komponente otkriva njenu namenu.



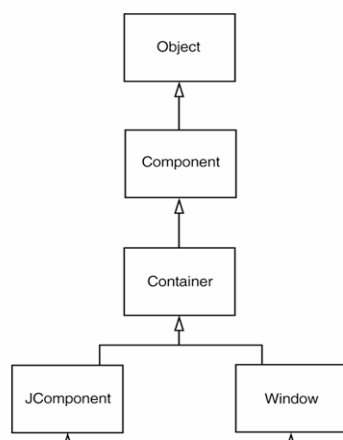
Svaka osoba koja je videla neku aplikaciju, za belu površinu očekuje da je polje za unos podataka. Zamislite da je to dugme u nekoj aplikaciji!!!

Swing takođe podržava „**Model-View-Controller**“ arhitekturu. Termin „Model-View-Controller“ predstavlja u softverskom inženjerstvu princip u kome se biznis logika(ponašanje) izoluje od korisničkog interfejsa. MVC pattern se koristi u aplikacijama gde je lakše menjati ili vizuelni izgled aplikacije ili njeno ponašanje bez efekta na ovu drugu osobinu.



Detalje MVC arhitekture i njena primena će biti obrađena prilikom rada sa JTable komponentom.

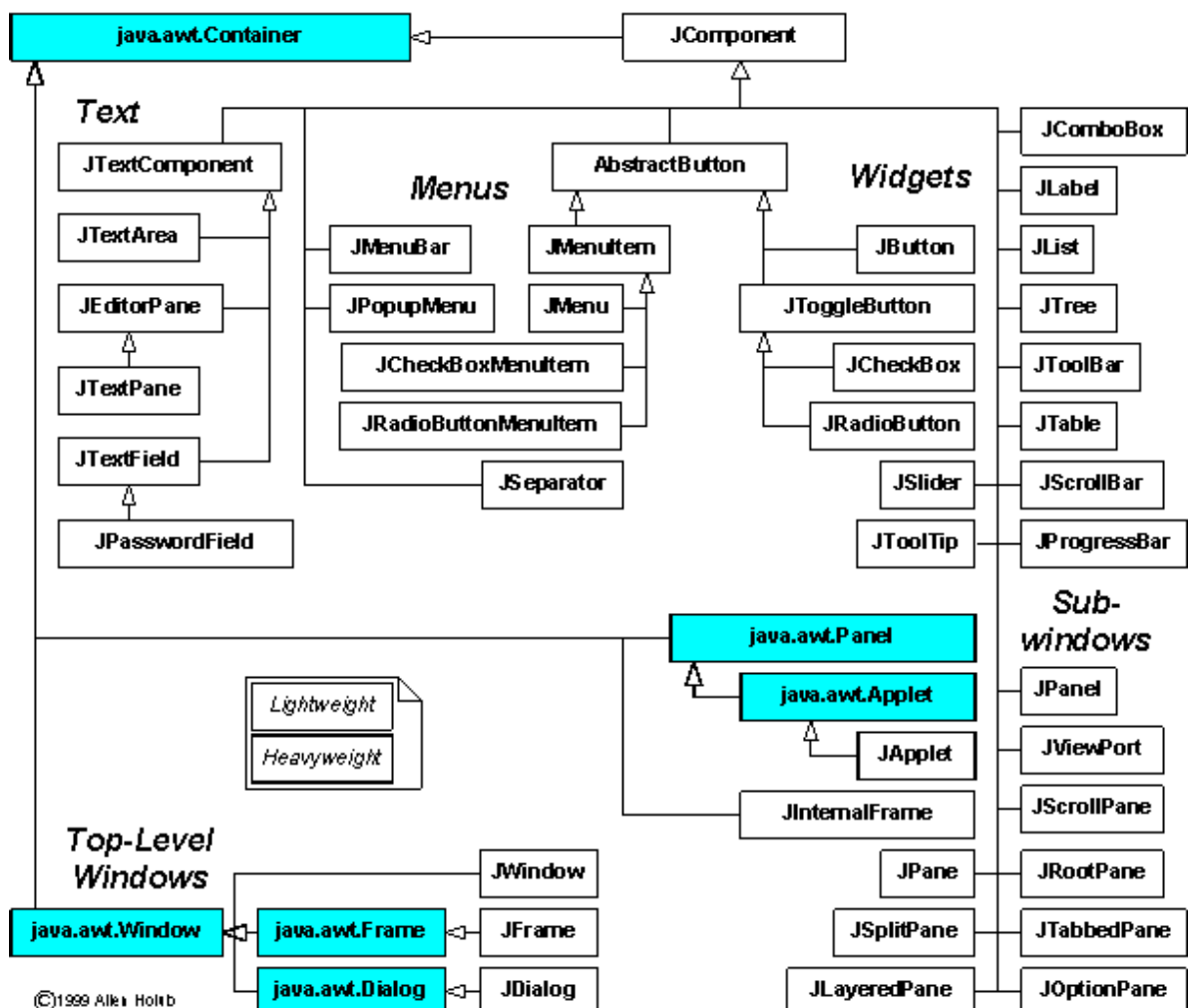
Delimična hijerarhija klasa AWT i Swing biblioteke



1. Sve Java klase nasleđuju klasu Object

2. Klasa `java.awt.Component` je nadklasa svih AWT GUI klasa

3. Klasa `javax.swing.JComponent` je nadklasa svih Swing GUI klasa. `javax.swing.JComponent` je podklasa klase `java.awt.Container` i indirektno je podklasa klase `java.awt.Component` preko klase `java.awt.Container`. To znači da u radu sa svim Swing komponentama mogu se koristiti metode klase: `java.awt.Component`, `java.awt.Container` i `javax.swing.JComponent`



Na slici je prikazana hijerarhija Swing biblioteke, odnosno njenih komponenti. Gotovo sve komponente će se koristiti u toku vežbi. Zajedničko za sve Swing komponente je da nazivi klase počinju sa Jxxxxxx.

3.2. Glavni prozor aplikacije - klasa JFrame

Frame predstavlja prozor najvišeg nivoa sa naslovom (title) i granicom (border). Frame implementiran kao instanca klase JFrame je prozor sa naslovom, granicom i dugmičima koji služe za zatvaranje prozora, minimiziranje i maksimiziranje prozora. Aplikacije koje koriste Swing GUI komponente obično imaju najmanje jedan Frame, odnosno imaju barem jednu instancu klase JFrame.

Pre kreiranja prve klase u našem projektu potrebno je kreirati projekat Termin3 i korenski paket (nazvati ga na primer gui, imena paketa uvek neka počinju malim slovom) unutar src foldera. Kreirati novu Java klasu: Desni klik na paket gui->New->Class. Klasi dati naziv MainFrame (klase će uvek počinjati velikim slovom). ***Ova klasa će nasleđivati klasu JFrame. Dakle u potpunosti nasleđuje izgled i ponašanje klase JFrame, sa svim atributima i metodama uz mogućnost redefinisanja istih.***

Klasa MainFrame.java:

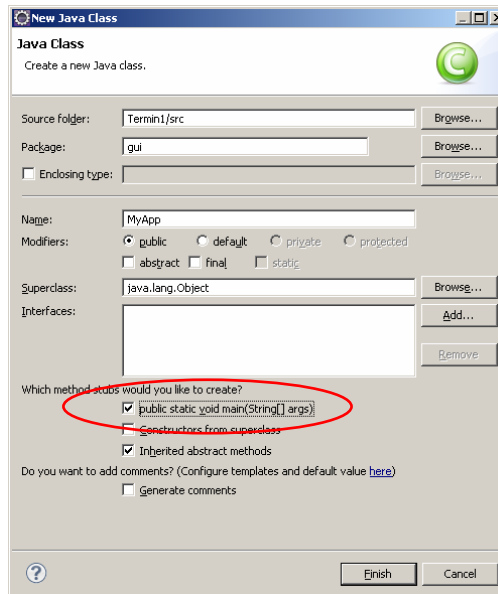
```
package gui;

import javax.swing.*;

public class MainFrame extends JFrame {
    public MainFrame() {
        super();
        setSize(300, 200); // Dimenzije su inicijalno (0, 0)
        setTitle("My First GUI App");
    }
}
```

1. Šta znači deklaracija package?
2. Šta znači deklaracija import?
3. Kako se u Eclipsu automatski uvode (importuju) potrebni paketi?
4. Šta predstavlja ključna reč extends?
5. Šta predstavlja metoda MainFrame()?
6. Šta predstavlja metoda super() ?
7. Metode setSize i setTitle predstavljaju metode koje klase?

Izvršavanje programa u Javi počinje sa `main()` metodom. Najlakši način u Eclipsu da kreiramo klasu koja sadrži `main()` metodu je: Desni klik na gui paket -> New -> Class



Obratiti pažnju da opcija „public static void main“ treba da bude čekirana. Na ovaj način novokreirana klasa nazvana „MyApp“ sadržaće metodu `main()` koja će biti početna tačka programa.

Unutar metode `main()` kreiramo jednu instancu klase `MainFrame` i prikazujemo je kao glavnu formu aplikacije.

Klasa `MyApp.java`

```
package gui;
```

```
public class MyApp {
```

```
    public static void main(String[] args) {
        MainFrame mf = new MainFrame();
        mf.setVisible(true); // Prozor je inicijalno nevidljiv
    }
```

```
}
```

Pokretanje Java aplikacije iz Eclipsa se izvodi na sledeći način:
Desni klik na klasu koja sadrži `main()` metodu -> Run As->Java Application

Posle nekoliko pokretanja aplikacije i zatvaranja iste otvoriti „Task Manager“ i proveriti u procesima koliko ima Java procesa (`javaw.exe`). Primetićete da je za svako pokretanje aplikacije pokrenut i novi Java proces ali da je isti i dalje aktivan bez obzira što je prozor glavne aplikacije zatvoren. Dakle zatvaranjem glavnog prozora aplikacija nije prestala sa radom već je samo glavni prozor postao „nevidljiv“.

Najvažnije metode klase JFrame:`javax.swing`**Class JFrame**

```

java.lang.Object
├── java.awt.Component
│   ├── java.awt.Container
│   │   ├── java.awt.Window
│   │   │   ├── java.awt.Frame
│   │   │   └── javax.swing.JFrame

```

Naziv metode	Matična klasa	Opis metode
<code>public boolean isVisible()</code>	<code>java.awt.Component</code>	Vraća true ili false u zavisnosti od toga da li je prozor vidljiv ili ne.
<code>public void setVisible(boolean b)</code>	<code>java.awt.Component</code>	Prikazuje ili sakriva prozor u zavisnosti od ulaznog parametra. Instanca klase <code>JFrame</code> je po default-u sakrivena, odnosno <code>visible = false</code> .
<code>public boolean isEnabled()</code>	<code>java.awt.Component</code>	Vraća true ili false u zavisnosti od toga da li je pristup prozoru omogućen.
<code>public void setEnabled(boolean b)</code>	<code>java.awt.Component</code>	Omogućava ili onemogućava prozor u zavisnosti od ulaznog parametra
<code>public void setCursor(Cursor cursor)</code>	<code>java.awt.Component</code>	Postavljanje izgleda kursora nad prozorom. Kao parametar se prosleđuje instanca klase <code>java.awt.Cursor</code>
<code>public Cursor getCursor()</code>	<code>java.awt.Component</code>	Vraća trenutno aktivan kursor nad instancom klase <code>JFrame</code>
<code>public void setBackground(Color c)</code>	<code>java.awt.Component</code>	Postavljanje boje pozadine prozora. Kao parametar se prosleđuje instanca klase <code>java.awt.Color</code>
<code>public Color getBackground()</code>	<code>java.awt.Component</code>	Vraća instancu klase <code>java.awt.Color</code> koja predstavlja trenutno aktivnu boju pozadine prozora.
<code>public void setLocation(int x,int y)</code>	<code>java.awt.Component</code>	Postavlja prozor na zadate koordinate u odnosu na levi gornji ugao ekrana
<code>public void setLocation(Point p)</code>	<code>java.awt.Component</code>	Postavlja prozor na zadate koordinate u odnosu na levi gornji ugao ekrana
<code>Point getLocation()</code>	<code>java.awt.Component</code>	Vraća trenutno aktivne koordinate prozora u odnosu na levi gornji ugao ekrana.

public void setSize(Dimension d)	java.awt.Component	Postavlja novu veličinu prozora
public void setSize(int width,int height)	java.awt.Component	Postavlja novu veličinu prozora
public Dimension getSize()	java.awt.Component	Vraća trenutno aktivnu dimanziju prozora.
public void toBack()	java.awt.Window	Sklanja prozor iza ostalih prozora
public void toFront()	java.awt.Window	Premešta prozor "u prvi plan" (ispred svih ostalih prozora)
public void pack()	java.awt.Window	Veličinu prozora automatski prilagođava veličini koja odgovara rasporedu komponenti koje se nalaze unutar prozora. Ako je forma prazna (ne sadrži nijednu komponentu) prozor će sadržati samo naslov i kontrolne dugmiće.
public Toolkit getToolkit()	java.awt.Window	Ova metoda vraća pomoćnu klasu Toolkit za glavni prozor. Ova klasa sadrži čitav niz interesantnih metoda za GUI i biće detaljnije obrađena u poglavlju 1.5.
public void setLocationRelativeTo(Component c)	java.awt.Window	Postavlja prozor u relativni položaj u odnosu na prosleđenu komponentu. Ukoliko je parametar NULL tada se prozor postavlja na centar ekrana.
public void setTitle(String title)	java.awt.Frame	Dodela naslova prozora
public String getTitle	java.awt.Frame	Dobijanje naslova prozora
public void setResizable(boolean resizable)	java.awt.Frame	Definiše da li korisnik može da menja veličinu prozora
public boolean isResizable()	java.awt.Frame	Metoda vraća indikator da li korisnik može da menja veličinu prozora
public void setUndecorated(boolean undecorated)	java.awt.Frame	Skida i vraća dugmad za minimizovanje i maksimizovanje prozora.
public boolean isUndecorated()	java.awt.Frame	Skida i vraća dugmad za minimizovanje i maksimizovanje prozora.
public int getExtendedState()	java.awt.Frame	Vraća stanje u kome se nalazi prozor. Moguća stanja su: <ul style="list-style-type: none"> • NORMAL • ICONIFIED • MAXIMIZED_HORIZ • MAXIMIZED_VERT • MAXIMIZED_BOTH Concatenates MAXIMIZED_HORIZ and

		MAXIMIZED_VERT.
public void setExtendedState(int state)	java.awt.Frame	Definiše novo stanje prozora. Moguća stanja u koja se prozor može postaviti su sledeća: <ul style="list-style-type: none"> • NORMAL • ICONIFIED • MAXIMIZED_HORIZ • MAXIMIZED_VERT • MAXIMIZED_BOTH
public void setMenuBar(MenuBar mb)	java.awt.Frame	Dodaje glavni meni na prozor
public void setIconImage(Image image)	javax.swing.JFrame	Postavlja ikonu prozora
public void setDefaultCloseOperation(int operation)	javax.swing.JFrame	definiše akciju prilikom zatvaranja prozora. Moguća vrednost parametra operation: <p>WindowConstants.DO_NOTHING_ON_CLOSE - nema operacije.</p> <p>WindowConstants.HIDE_ON_CLOSE - prozor čini nevidljivim</p> <p>WindowConstants.DISPOSE_ON_CLOSE - sakriva i uništava prozor</p> <p>WindowConstants.EXIT_ON_CLOSE - izlazi iz aplikacije pozivajući metodu <code>exit()</code>, Ova operacija se koristi samo u aplikacijama</p>

Kako svaka klasa iz paketa `javax.swing` ima veći broj nadklasa, u svakoj swing klasi dostupan je veliki broj metoda koje ne vredi ni pokušati zapamtiti napamet. Najbolje je pri radu sa nekom klasom pogledati informativno dostupnu on-line dokumentaciju i okvirno upoznati mogućnosti klase: <http://lamp.epfl.ch/java/jdk1.6/docs/api/javax/swing/JFrame.html> a tek po potrebi detaljno razmatrati metode klase i tražiti metode koje odražuju potreban zadatak.

3.3. Klasa java.awt.Toolkit

Ova klasa sadrži metode koje omogućavaju kreiranje GUI komponenti iz paketa java.awt i javax.swing nezavisno od platforme na kojoj se Java nalazi (Windows, Linux...). Određeni broj metoda ne treba direktno pozivati (createXXXXX), dok jedan deo metoda daje informacije o vizuelnom okruženju u kome se aplikacije izvršava. Metode klase java.awt.Toolkit: <http://lamp.epfl.ch/java/jdk1.6/docs/api/java/awt/Toolkit.html>

Naziv metode	Opis metoda
public abstract void beep()	Emituje zvuk beep.
public abstract Image getImage(String filename)	Vraća sliku iz fajla čija je putanja navedena u ulaznom parametru. Veličina slike odgovara veličini slike u fajlu u pikselima. Podržani formati su: .GIF, .JPEG, .PNG.
public abstract int getScreenResolution()	Vraća tekuću rezoluciju ekrana u tačkama po inču.
public abstract Dimension getScreenSize()	Vraća tekuće dimanzije ekrana

Primer 1 . Aplikaciju koju smo kreirali ćemo izmeniti na sledeći način: Kreirati folder "images" u korenu projekta "Termin3" (desni klik nad projektom Termin3-> New -> Folder) i snimiti u ovaj folder sliku "iko.jpg" (bilo koja slika). Izmeniti klasu MainFrame na sledeći način:

```
package gui;
```

```
import java.awt.Dimension;
```

```
import java.awt.Image;
```

```
import java.awt.Toolkit;
```

```
import javax.swing.JFrame;
```

```
public class MainFrame extends JFrame {
```

```
    public MainFrame() {
```

```
        Toolkit kit = Toolkit.getDefaultToolkit();
```

```
        Dimension screenSize = kit.getScreenSize();
```

```
        int screenHeight = screenSize.height;
```

```
        int screenWidth = screenSize.width;
```

```
        // Podesavamo dimenzije prozora na polovinu dimenzija ekrana
```

```
        setSize(screenWidth / 2, screenHeight / 2);
```

```
        // Dodeljujemo ikonu
```

```
        Image img = kit.getImage("images/iko.jpg");
```

```
        setIconImage(img);
```

```
        //Podesavamo naslov
```

```
        setTitle("My Second GUI App");
```

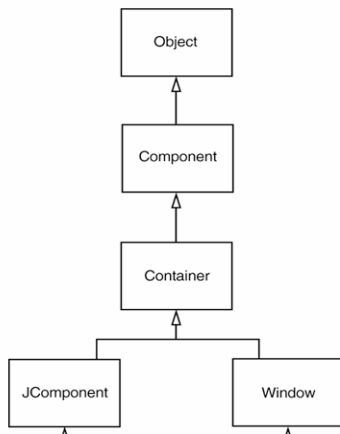
```
        //Postavljamo akciju pri zatvaranju prozora
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    }
```

```
}
```

3.4. Povezivanje JFC-a sa Java izvornim kodom



Hijerarhija klasa unutar swing paketa dovodi do toga da se u radu sa svim Swing komponentama mogu koristiti metode klasa: `java.awt.Component`, `java.awt.Container` i `javax.swing.JComponent`, odnosno svaka klasa poseduje veliki broj metoda što često može da deluje nepregledno i zbunjujuće.

Značajnu pomoć u radu pruža mogućnost uvida u izvorni kod Java klasa koje se nalazi u JFC-u. Da bi se izvorni kod povezao sa kompajliranim Java klasama koje se koriste u Eclipsu potrebno je odraditi sledeće:

1. Selektovati jedan projekat u Package Exploreru, desni klik-> Properties
2. Odabrati stavku Java Build Path
3. Odabrati karticu Libraries
4. Pronaći biblioteku `rt.jar` i u okviru nje stavku `source attachment`
5. Odabrati Putanja kao „External File“ i pronaći arhivu `src.zip` (nalazi se u `jdk` direktorijumu, ukoliko se koristi jre postoji mogućnost zasebnog downloada)

Nakon povezivanja JFC klasa sa Java izvornim kodom postoji mogućnost kompletnog pregleda svih Javinih klasa direktno iz Eclipse radnog okruženje (CTRL + levi klik na bilo koju Java klasu ili metodu).

3.5. Primer složene GUI aplikacije

Kao primer složene Java GUI aplikacije biće prikazan grafički editor Use-Case dijagrama. Importovati i pokrenuti projekat `GraphicEditor`.

3.6. Zadaci

1. Kreirati glavni prozor aplikacije koji je jednak četvrtini ekrana po dužini i širini
2. Glavni prozor aplikacije postaviti na centar ekrana
3. Promeniti boju pozadine glavnog prozora u žutu boju.