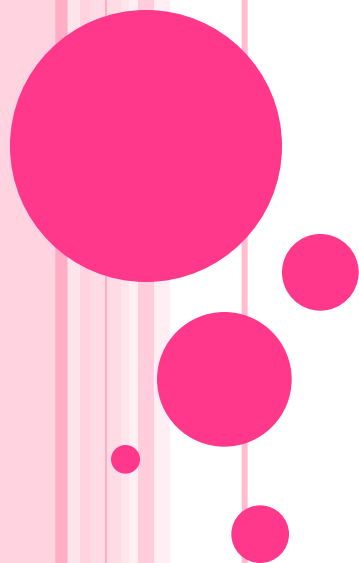




11

OBJEKTNO ORIJENTISANO PROGRAMIRANJE VEŽBE

Staša Vujičić Stanković





GUI U JAVI

- GUI – Graphical User Interface omogućava interfejs između korisnika i aplikacije koji je baziran na grafičkim komponentama.
- Ranije su se osnovni elementi potrebni za kreiranje GUI-a nalazili u paketima java.awt i javax.swing.
- awt je skraćenica od Abstract Window Toolkit. Ovaj paket je bio osnovni paket za kreiranje GUI-a do pojave javax.swing paketa sa Java 2 verzijom.
- Većina klasa iz paketa java.awt je zamenjena klasama iz paketa javax.swing.



GUI U JAVI

- Najveći broj klasa iz paketa javax.swing definiše GUI elemente, poznate kao **Swing komponente**, koje daju unapređene alternative komponentama definisanim u java.awt.
- Ipak, klase iz paketa java.awt ne mogu se ignorisati, jer javax.swing klase su iz njih izvedene i od njih zavise.



GUI U JAVI

- Swing klase su deo skupa GUI programskih mogućnosti koje nazivamo **JFC (Java Foundation Classes)**.

Tu spadaju:

- Swing klase
- klase za 2D crtanje iz paketa `java.awt.geom`
- klase koje podržavaju drag-and-drop iz paketa `java.awt.dnd`
- API (Application Program Interface) iz paketa `javax.accessibility` koji omogućava implemeniranje aplikacija.



JAVAFX

- JavaFX , čist Java API, je sledeća generacija GUI alata za razvoj (RIA – rich Internet applications) aplikacija.
- JavaFX je Java biblioteka predviđena za upotrebu na različitim tipovima uređaja kao što su mobilni telefoni, pametni telefoni, računari, tablet računari itd.
i uključuje rad sa audio i video sadržajem, animacijom i grafikom.





HELLOWORLDMAIN.JAVA

Za rad sa klasama paketa javafx neophodno je u putanju (Build Path...) dodati biblioteku JavaFX SDK (jfxrt.jar – <JavaFX SDK Home directory>\rt\lib\jfxrt.jar.)

```
import javafx.application.Application;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.scene.Group;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.stage.Stage;
```



JavaFX aplikacije nasleđuju klasu **javafx.application.Application** koja omogućava životni ciklus aplikacijama, kroz funkcije kao što su pokretanje i zaustavljanje. Ova klasa takođe omogućava bezbedno pokretanje GUI komponenti u višenitnom okruženju.

```
public class HelloWorldMain extends Application
{
    public static void main(String[] args)
    {
        Application.launch(args);
    }
}
```



- U metodu **main()** poziva se metod **Application.launch()** i prosleđuju mu se eventualni argumenti komandne linije.
- Nakon pokretanja aplikacije, interno se vrši poziv metoda **start()** nakon čega je programeru na raspolaganju JavaFX objekat **Stage** - **javafx.stage.Stage** za dalju upotrebu i manipulaciju.



- Stvaraoci JavaFX API-ja su organizovali prikaz grafičkog korisničkog interfejsa slično pozorištu, gde se predstava igra na bini ispred publike.
- Zbog toga se na jednoj bini (**Stage**), može postaviti veći broj scena
Scene - javafx.scene.Scene, na kojima se odigravaju „predstave“.
- Objekat Stage iz JavaFX je ekvivalentan prozorima aplikacije (JFrame ili JDialog) koji su se koristili u Java Swing API.



- Objekat **Scene** može da se posmatra kao okvir za sadržaj (content pane) u kome može da bude smešten proizvoljan broj čvorova – objekata tipa **Node - javafx.scene.Node**.
- Node je osnovna klasa čvorova grafa scene. Najčešće se koriste čvorovi kontrole ulaza/izlaza ili Shape objekti.
- Graf scene je drvolika struktura koja kao čvorove ima jednog roditelja i decu, koji su organizovani korišćenjem kontejnerske klase Group.
- Nakon dodavanja čvorova, scena se postavlja na binu i poziva se metod **show()** za prikaz JavaFX prozora.



JAVAFX.SCENE.NODE

- Bazna klasa za predstavljanje čvorova grafa scene.
- Graf scene je drvoliki skup podataka u kome svaki čvor ima nula ili jednog roditelja , i svaki čvor je ili „list“ sa nula dece ili „grana“ sa nula ili više dece.
- U svakom drvetu koje predstavlja graf scene postoji samo jedan čvor koji nema roditelja. On se naziva čvor koren – „**root**“.



JAVAFX.SCENE.NODE

- Svaki čvor je tipa Node ili neke njene podklase.

Čvorovi-grane su specifično tipa **javafx.scene.Parent** čije su podklase **Group**, **Region** i **Control**.

Čvorovi-listovi su specifično tipa **Rectangle**, **Text**, **ImageView**, **MediaView**, ili bilo koje klase koja ne može da ima dece.

- <http://docs.oracle.com/javafx/2/api/javafx/scene/Node.html>



JAVAFX.SCENE.PARENT

- Bazna klasa za sve čvorove grafa scene koji mogu da imaju decu.
- Ova klasa upravlja svim hijerarhijskim operacijama grafa scene, uključujući dodavanje/uklanjanje dece čvorova itd.
- Ima tri direktne podklase:
 - **Group** – efekti i transformacije koji će da se primenjuju na kolekciju čvorova dece.
 - **Region** – površina ekrana koja može da sadrži i druge čvorove i da bude stilizovana korišćenjem CSS-a.
 - **Control** – klasa na visokom nivou za čvorove namenjene za interakciju korisnika.
- <http://docs.oracle.com/javafx/2/api/javafx/scene/Parent.html>



```
public void start(Stage primaryStage)
{
    primaryStage.setTitle("Hello World");
    Group root = new Group();
    Scene scene = new Scene(root, 300, 250);
    Button btn = new Button();
    btn.setLayoutX(100);
    btn.setLayoutY(80);
    btn.setText("Hello World");
    btn.setOnAction(new EventHandler<ActionEvent>() {
        public void handle(ActionEvent event)
        {
            System.out.println("Hello World");
        }
    });
    root.getChildren().add(btn);
    primaryStage.setScene(scene);
    primaryStage.show();
}
```



JAVAFX.SCENE.PAINT.COLOR

- Standardni konstruktor:

**Color(double red, double green,
double blue, double opacity)**

red, green, blue, opacity
pripadaju intervalu 0-1.0.



JAVAFX.SCENE.PAINT.COLOR

- Boja se definiše kao kombinacija tri osnovne boje: crvene, zelene i plave.
- To su tzv. **RGB** vrednosti za boje.
- Svaka osnovna boja može imati vrednost od 0 do 255.
- Color boja = Color.rgb(0,0,0); - crna
Color boja = Color.rgb(255,255,255); - bela



JAVAFX.SCENE.PAINT.COLOR

- U HSB modelu:

**hsb(double hue, double saturation,
double brightness)**

**hsb(double hue, double saturation,
double brightness, double opacity)**
(nijansa, zasićenost, osvetljenje, neprozirnost)

- **Color boja = Color.hsb(270,1.0,1.0);**
Color boja = Color.hsb(270,1.0,1.0,1.0);



JAVAFX.SCENE.PAINT.COLOR

- Svaka boja ima implicitnu alpha vrednost 1.0 ili eksplicitnu vrednost zadatu u konstruktoru.
- Alpha vrednost definiše transparentnost boje i može da se predstavi dvostrukim vrednosnim opsezima 0.0-1.0 ili 0-255.
- Alpha vrednost 1.0 ili 255 znači da je boja potpuno neprozirna, dok 0.0 ili 0 vrednosti označavaju potpuno transparentnu boju.
- `Color boja = new Color(0,0,1,0.7); // alpha = 0.7`



JAVAFX.SCENE.PAINT.COLOR

- Postoji mogućnost da se RGB boja odredi HTML ili CSS string atributom:

web(java.lang.String colorString)

web(java.lang.String colorString, double opacity)

- **Color boja = Color.web("0x0000FF",1.0);**
// plavo hex web vrednost, eksplicitno zadato alpha



JAVAFX.SCENE.PAINT.COLOR

- Klasa *Color* definiše i *public final static* promenljive za standardne boje:

WHITE (255,255,255)

RED (255,0,0)

PINK (255,175,175)

LIGHTGRAY (192,192,192)

ORANGE (255,200,0)

MAGENTA (255,0,255)

GRAY (128,128,128)

YELLOW (255,255,0)

CYAN (0,255,255)

DARKGRAY (64,64,64)

GREEN (0,255,0)

BLUE (0,0,255)

BLACK (0,0,0)



JAVAFX.SCENE.PAINT.COLOR

- Kada smo kreirali Color objekat, možemo posvetliti ili potamniti boju pozivom metoda **brighter()** i **darker()** koji uvećavaju ili umanjuju intenzitet komponenti boje za predefinisani faktor

```
boja.brighter();  
boja.darker();
```



JAVAFX.SCENE.PAINT.COLOR

- Metode:

getRed(), getGreen(), getBlue()

– vraćaju komponente boje

equals()

– za poređenje boja na jednakost komponenti

...

- <http://docs.oracle.com/javafx/2/api/javafx/scene/paint/Color.html>



JAVAFX.SCENE.CURSОР

- Klasa Cursor definiše static konstante koje određuju standardne tipove kursora. Npr.

CLOSED_HAND
DEFAULT
TEXT

CROSSHAIR
HAND
WAIT

- <http://docs.oracle.com/javafx/2/api/javafx/scene/Cursor.html>



JAVA.AWT.FONT

- Klasa Font je dosta komplikovana, navodimo samo najbitnija svojstva.
- Klasa Font pravi razliku između karaktera i glifa, tj. grafičke reprezentacije karaktera. Različiti fontovi definišu različite glifove za jedan isti karakter.
- Konstruktori:

Font(double size)

Font zadate veličine predefinisano „System“ tipa.

Font(java.lang.String name, double size)

Font zadate veličine zadatog tipa.



JAVA.AWT.FONT

- Metode:

getStyle() – vraća stil fonta

getSize() – vraća veličinu fonta

font(java.lang.String family, double size)

font(java.lang.String family, FontPosture posture, double size)

...

- <http://docs.oracle.com/javafx/2/api/javafx/scene/text/Font.html>



LAYOUT PANES

- Layout pane za scenu određuje poziciju i veličinu svih čvorova na njoj i njime se zadaje izgled grafičkog korisničkog interfejsa.
- Kako se menja veličina prozora, layout pane automatski odgovara i menja veličinu čvorova koje sadrži u skladu sa njihovim osobinama.



LAYOUT PANES

- JavaFX sadrži niz layout klasa, koje se nalaze u paketu: **javafx.scene.layout**. To su klase:
- **AnchorPane**
- **BorderPane**
- **StackPane**
- **HBox**
- **VBox**
- **TilePane**
- **FlowPane**
- **GridPane**



LAYOUT PANES

- **AnchorPane** – Čvorovi deca se sidre po ivicama roditelja.
Veličina čvorova dece ne može da se menja.
- **BorderPane** – Površina roditelja se deli na 5 pozicija – vrh, levo, desno, dno, centar.
Veličina čvorova dece smeštenih na pozicije vrh i dno se menjaju po horizontali, onih koji su smešteni na pozicije levo i desno po vertikali, a onih koji su smešteni u centru po oba.
Sva povećanja veličina vrše se do maksimalne veličine čvora u relevantnom smeru.



LAYOUT PANES

- **StackPane** – Čvorovi deca se raspoređuju jedna iznad druge, kao špil karata.
Jedino je komponenta na “vrhu” vidljiva u bilo kom trenutku.
Veličina čvorova dece se menja, tako da ispuni veličinu roditelja, uz poštovanje zadate maksimalne širine, odnosno visine za svaki čvor.



LAYOUT PANES

- **HBox** – Čvorovi deca se raspoređuju u jedan red. Veličina čvorova dece se povećava do njihove željene širine (*preferred widths*), ali postoji mogućnost da se zada da se pojedini čvorovi deca povećavaju do njihove maksimalne zadate širine.
- **VBox** – sličan HBox-u, jedino što se čvorovi deca raspoređuju u jednu kolonu.

Padding property upravlja rastojanjem između čvorova. *Margine* takođe mogu da se podese da bi se odredila veličina praznog prostora oko pojedinih čvorova.



LAYOUT PANES

- **FlowPane** – dodaju se komponente u sukcesivnim redovima – kad je red popunjen, počinje se sa novim. Najčešće se koristi za uređivanje dugmeta. Veličina čvorova dece ne može da se menja.
- **TilePane** – Sličan kao FlowPane, jedino što se čvorovi deca smeštaju u mrežu u kojoj je svaka „pločica“ jednake veličine. Čvorovi deca mogu da budu postavljeni horizontalno (u redovima) ili vertikalno (u kolonama). Veličina čvorova dece menja se, tako da ispuni veličinu „pločice“, uz poštovanje zadate maksimalne širine i visine.



LAYOUT PANES

- **GridPane** – komponente se raspoređuju u fleksibilnu pravougaonu mrežu. Promena veličine zavisi od ograničenja zadatih za svaki red, odnosno kolonu, pojedinačno.

FLOWPANE

- FlowPane smešta čvorove decu u red, a kada se red napuni, automatski počinje smeštanje u novi red.
- Podrazumevana pozicija reda sa komponentama je centar kontejnera, a podrazumevana orijentacija je s leva na desno.
- Može se zadati podrazumevani razmak među komponentama.
- <http://docs.oracle.com/javafx/2/api/javafx/scene/layout/FlowPane.html>





FLOWPANE - KONSTRUKTORI

- **FlowPane()** – kreira horizontalni FlowPane sa horizontalnim i vertikalnim razmakom 0 ($hgap/vgap = 0$).
- **FlowPane(double hgap, double vgap)** – kreira horizontalni FlowPane sa zadatim hgap i vgap.
- **FlowPane(Orientation orientation)** – kreira FlowPane zadate orijentacije sa horizontalnim i vertikalnim razmakom 0 ($hgap/vgap = 0$)
- **FlowPane(Orientation orientation, double hgap, double vgap)** – kreira FlowPane zadate orijentacije sa zadatim hgap i vgap.



FLOWPANE

- Primer horizontalnog FlowPane-a:

```
Image images[] = { ... };
```

```
FlowPane flow = new FlowPane();
```

```
flow.setVgap(8);
```

```
flow.setHgap(4);
```

```
flow.setPrefWrapLength(300); // preferred width = 300
```

```
for (int i = 0; i < images.length; i++) {
```

```
    flow.getChildren().add(new ImageView(image[i]));
```

```
}
```



FLOWPANE

- Primer vertikalnog FlowPane-a:

```
FlowPane flow = new
    FlowPane(Orientation.VERTICAL);

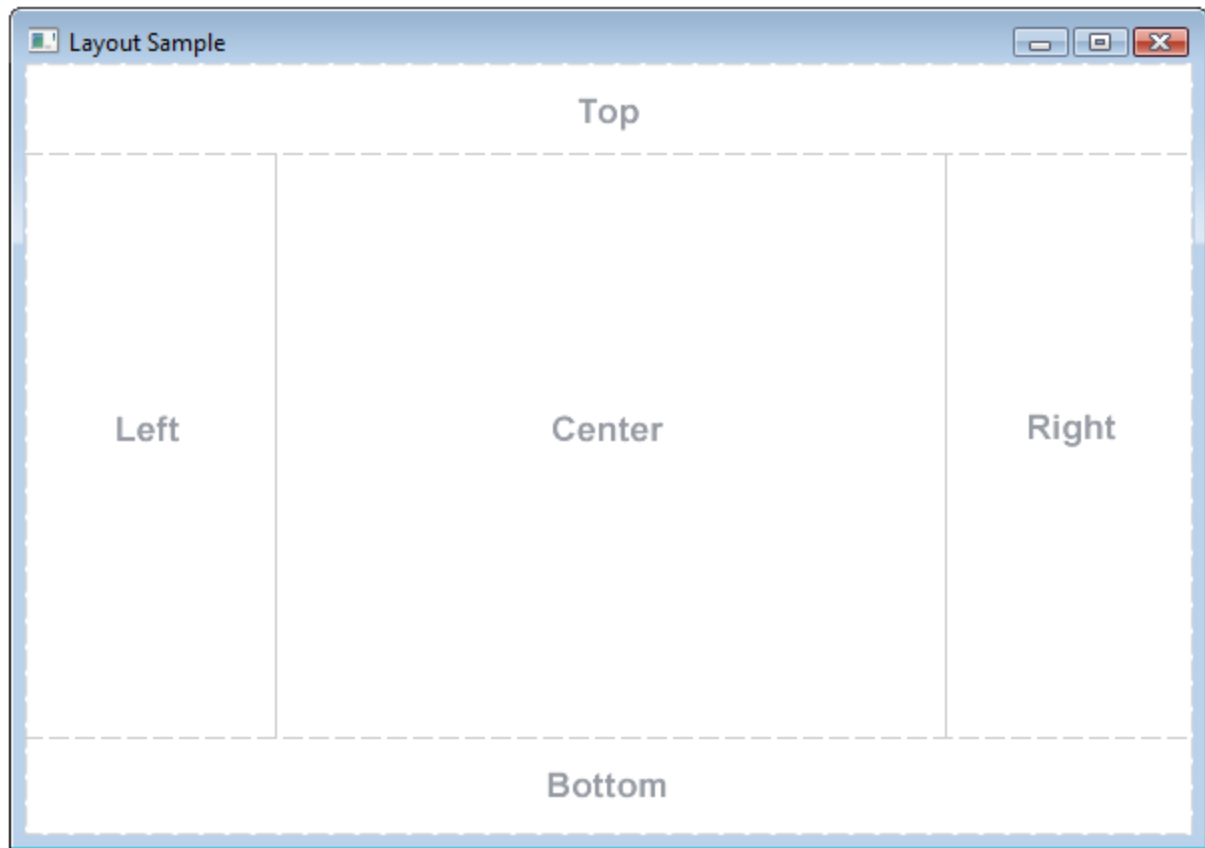
flow.setColumnHalignment(HPos.LEFT);
// poravnanje elemenata: levo

flow.setPrefWrapLength(200);
// željena visina = 200

for (int i = 0; i < titles.size(); i++) {
    flow.getChildren().add(new Label(titles[i]));
}
```

BORDERPANE

- Komponente raspoređene u BorderPane se šire tako da popune raspoloživ prostor u kontejneru.
- Bilo koja pozicija može da bude null.
- Pozadina i ivice BorderPane-a mogu da budu stilizovane CSS-om.
- <http://docs.oracle.com/javafx/2/api/javafx/scene/layout/BorderPane.html>





BORDERPANE

- Konstruktor: **BorderPane()**.
- Primer:

```
BorderPane borderpane = new BorderPane();
```

```
ToolBar toolbar = new ToolBar();
```

```
HBox statusbar = new HBox();
```

```
Node appContent = new AppContentNode();
```

```
borderPane.setTop(toolbar);
```

```
borderPane.setCenter(appContent);
```

```
borderPane.setBottom(statusbar);
```



GRIDPANE

- GridPane raspoređuje čvorove u fleksibilnu pravougaonu mrežu.

	Sales: Current Year		
	Goods and Services		
			Services 20%
	Goods 80%		

Čvorovi mogu da budu smešteni u bilo koju ćeliju u mreži. Mreža je korisna za upotrebu pri potrebi za raspoređivanje čvorova u redove i kolone.

- <http://docs.oracle.com/javafx/2/api/javafx/scene/layout/GridPane.html>



GRIDPANE

- Konstruktor: **GridPane()**.
- Primer:

```
GridPane gridpane = new GridPane();
```

```
// Postavljanje jedne po jedne osobine...
```

```
Button button = new Button();
```

```
GridPane.setRowIndex(button, 1);
```

```
GridPane.setColumnIndex(button, 2);
```





```
// Postavljanje svih odjednom  
Label label = new Label();
```

```
GridPane.setConstraints(label, 3, 1);  
// kolona=3 red=1
```

```
// obavezno je postaviti decu na GridPane  
gridpane.getChildren().addAll(button, label);
```



- Moguće je ujedno odrediti pozicije i postaviti decu:

```
GridPane gridpane = new GridPane();
```

```
gridpane.add(new Button(), 2, 1);
```

```
// kolona=2 red=1
```

```
gridpane.add(new Label(), 3, 1);
```

```
// kolona=3 red=1
```



PAKET JAVAFX.GEOMETRY

CLASS HIERARCHY

- Obezbeđuje skup 2D klasa za definisanje i obavljanje operacija nad objektima u dvodimenzionoj geometriji.
- javafx.geometry.**BoundingBoxBuilder**
(implements javafx.util.Builder<T>)
- javafx.geometry.**Bounds**
- javafx.geometry.**BoundingBox**
- javafx.geometry.**Dimension2D**
- javafx.geometry.**Dimension2DBuilder**
(implements javafx.util.Builder<T>)
- javafx.geometry.**Insets**
- javafx.geometry.**InsetsBuilder**
(implements javafx.util.Builder<T>)





- `javafx.geometry.Point2D`
- `javafx.geometry.Point2DBuilder<B>`
(implements `javafx.util.Builder<T>`)
- `javafx.geometry.Point3D`
- `javafx.geometry.Point3DBuilder<B>`
(implements `javafx.util.Builder<T>`)
- `javafx.geometry.Rectangle2D`
- `javafx.geometry.Rectangle2DBuilder<B>`
(implements `javafx.util.Builder<T>`)



PAKET JAVAFX.GEOMETRY

ENUM HIERARCHY

- javafx.geometry.**VPos**
 - javafx.geometry.**VerticalDirection**
 - javafx.geometry.**Side**
 - javafx.geometry.**Pos**
 - javafx.geometry.**Orientation**
 - javafx.geometry.**HPos**
 - javafx.geometry.**HorizontalDirection**
-
- <http://docs.oracle.com/javafx/2/api/javafx/geometry/package-summary.html>



JAVAFX.GEOMETRY.POS

- Skup vrednosti za opisivanje vertikalnog i horizontalnog pozicioniranja i poravnanja.
- BASELINE_CENTER, BASELINE_LEFT, BASELINE_RIGHT
- BOTTOM_CENTER, BOTTOM_LEFT, BOTTOM_RIGHT
- CENTER, CENTER_LEFT, CENTER_RIGHT
- TOP_CENTER, TOP_LEFT, TOP_RIGHT
- <http://docs.oracle.com/javafx/2/api/javafx/geometry/Pos.html>



JAVAFX.GEOMETRY.POS

- Metode:

HPos getHpos()

– vraća horizontalnu poziciju/poravnanje

VPos getVpos()

– vraća vertikalnu poziciju/poravnanje

static Pos valueOf(java.lang.String name)

– vraća enum konstantu zadatog imenom name

static Pos[] values()

– vraća niz konstanti enum tipa, redom kojim su deklarirani.



JAVAFX.GEOMETRY.HPOS

- Skup vrednosti za opisivanje horizontalnog pozicioniranja i poravnavanja.
- CENTER
- LEFT
- RIGHT
- **static HPos valueOf(java.lang.String name)**
- **static HPos[] values()**
- <http://docs.oracle.com/javafx/2/api/javafx/geometry/HPos.html>



JAVAFX.GEOMETRY.VPos

- Skup vrednosti za opisivanje vertikalnog pozicioniranja i poravnavanja.
- BASELINE
- BOTTOM
- CENTER
- TOP
- **static VPos valueOf(java.lang.String name)**
- **static VPos[] values()**
- <http://docs.oracle.com/javafx/2/api/javafx/geometry/VPos.html>



JAVAFX.GEOMETRY.ORIENTATION

- Određivanje horizontalne i vertikalne orijentacije.
- HORIZONTAL
 - Horizontalna (levo $\leftrightarrow$ desno) orijentacija.
- VERTICAL
 - Vertikalna (vrh $\leftrightarrow$ dno) orijentacija.
- **static Orientation valueOf(java.lang.String name)**
- **static Orientation[] values()**
- <http://docs.oracle.com/javafx/2/api/javafx/geometry/Orientation.html>



JAVAFX.GEOMETRY.INSETS

- Skup rastojanja za 4 strane pravougaone površine.
- Konstruktori:

Insets(double topRightBottomLeft)

Formira se Inset objekat koji ima jednaku vrednost rastojanja za sva četiri razmaka.

Insets(double top, double right, double bottom, double left)

Zadaju se 4 različite vrednosti za rastojanja.

- <http://docs.oracle.com/javafx/2/api/javafx/geometry/Insets.html>