



Programiranje korisničkih interfejsa 2018/19

Desktop interfejs

Sadržaj

- Instalacija alata
- Kreiranje HelloWorldApp projekta
- Layout Panes
- Meniji
- CheckBox i RadioButton
- Input boxes
- Promena scene
- Prozori

Instalacija alata

- Tehnologija: JavaFX

- Platforma za dizajniranje grafičkih desktop i web aplikacija.
- Dolazi kao biblioteka (jfxrt.jar) uz Java Development Kit.
- Može se koristiti u bilo kom Java Integrated Development Environment –u za programiranje u Javi

- Preuzimanje alata:

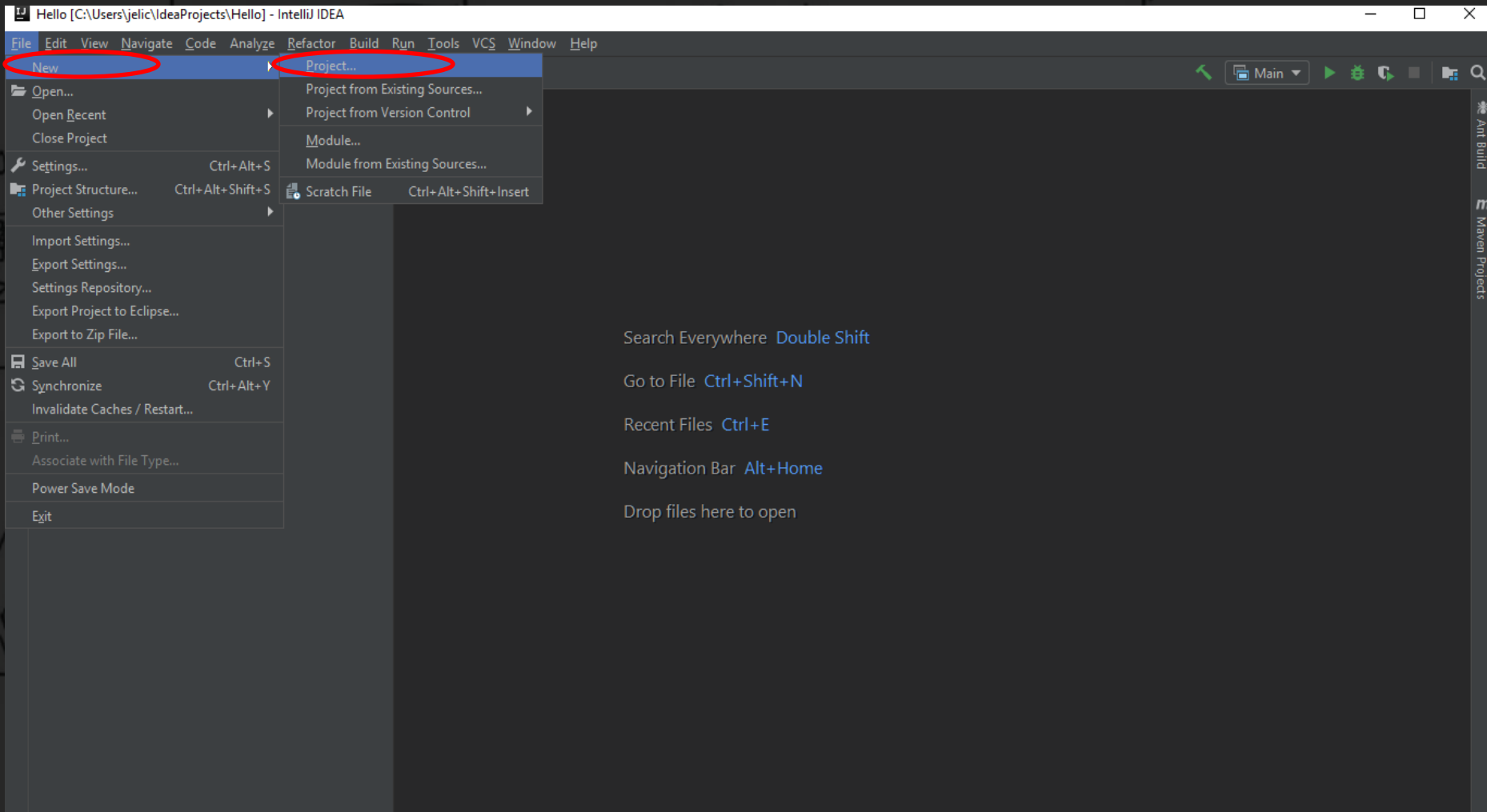
- <http://www.oracle.com/technetwork/java/javase/downloads/jdk9-downloads-3848520.html> - jdk9 (novembar 2017)
- <https://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/keplersr2> - Eclipse IDE (novembar 2017)
- <https://netbeans.org/downloads/> - NetBeans IDE (novembar 2017)
- <https://www.jetbrains.com/idea/download/#section=windows> - IntelliJ IDEA IDE (novembar 2017)
- Dostupni za sve platforme Windows/Linux/Mac

- Dokumentacija:

- <https://docs.oracle.com/javase/8/javafx/api/toc.htm> (novembar 2017)

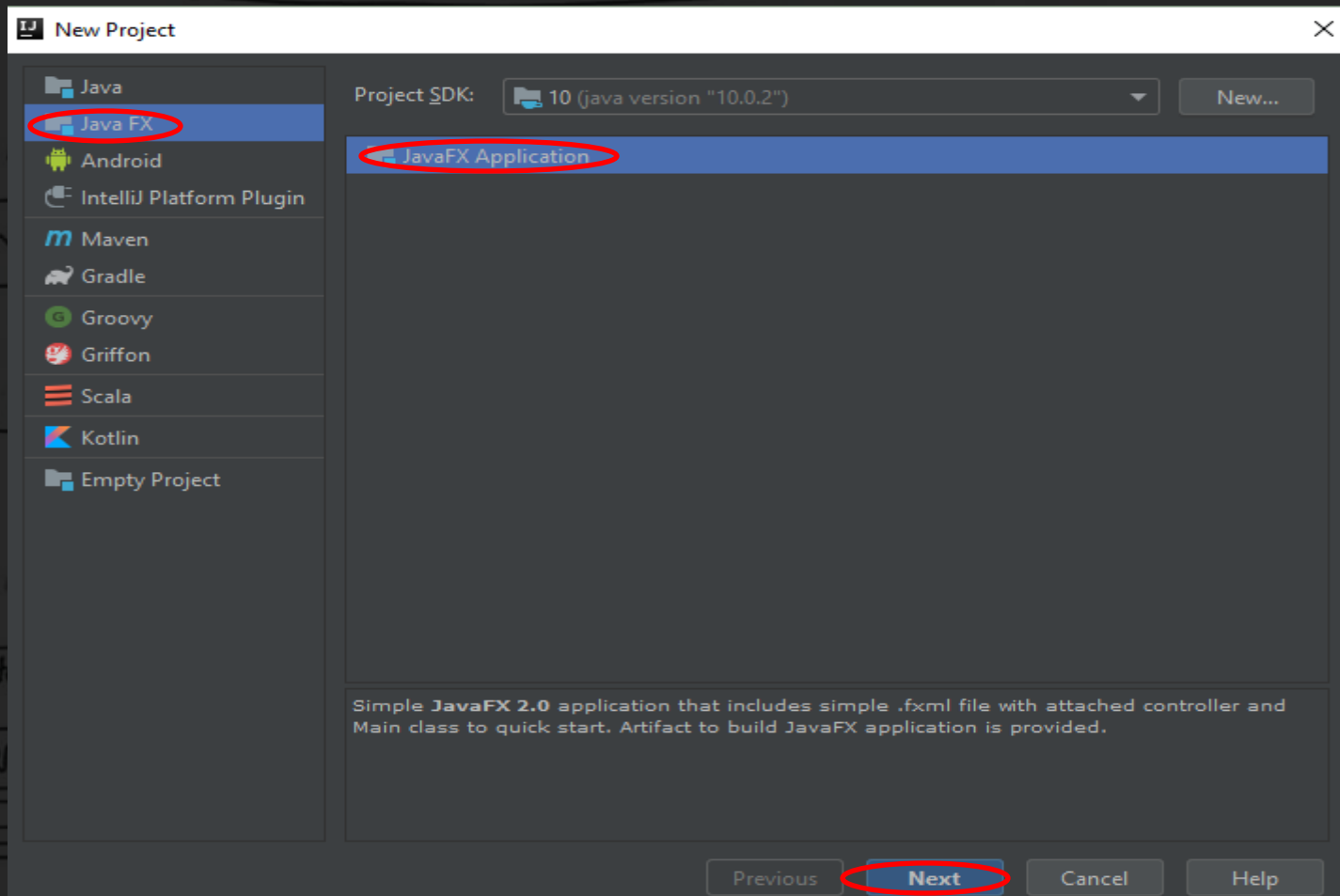
Kreiranje HelloWorldApp projekta

- 1) Kreiranje JavaFX projekta u IntelliJ-u
 - i. Napraviti novi projekat File->New->Project



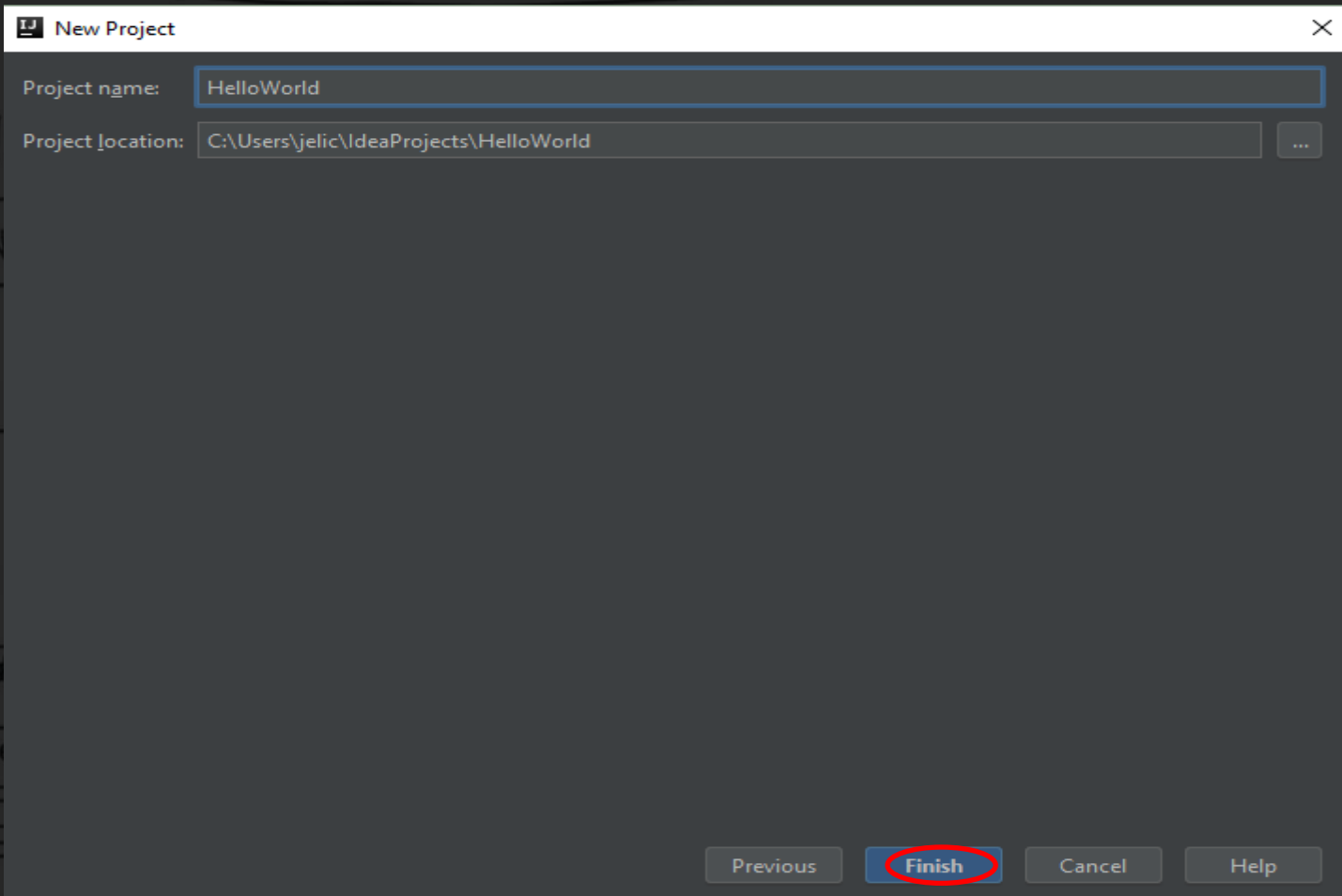
Kreiranje HelloWorldApp projekta

- 1) Kreiranje JavaFX projekta u IntelliJ-u
 - i. Izabрати JavaFx Application



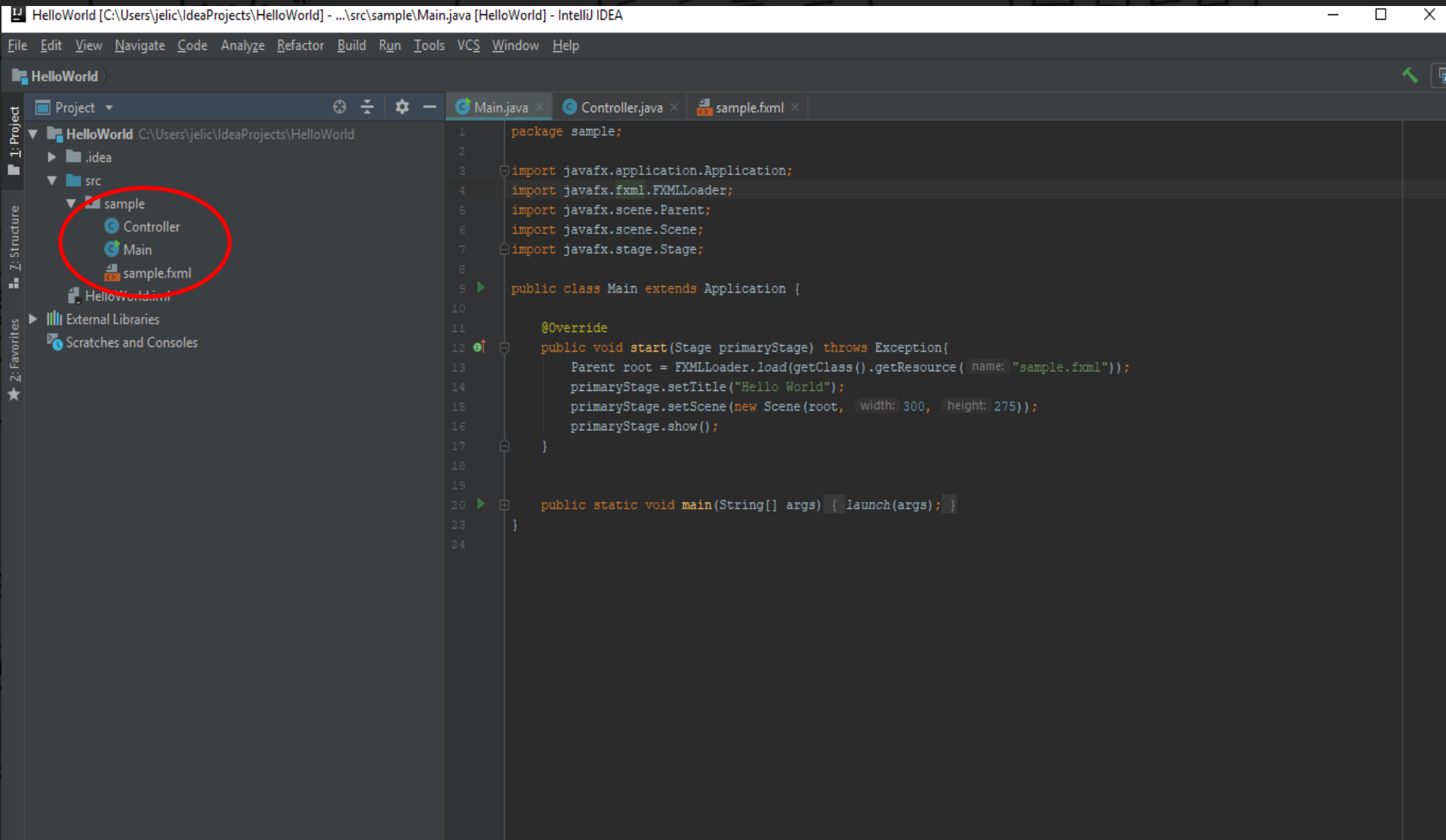
Kreiranje HelloWorldApp projekta

- 1) Kreiranje JavaFX projekta u IntelliJ-u
 - i. Uneti naziv projekta i njegovu lokaciju



Kreiranje HelloWorldApp projekta

2) Osnovni elementi JavaFX aplikacije – MVC pattern



Kreiranje HelloWorldApp projekta

2) Osnovni elementi JavaFX aplikacije

```
1 package sample;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.stage.Stage;
8
9 public class Main extends Application {
10
11     @Override
12     public void start(Stage primaryStage) throws Exception{
13         Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));
14         primaryStage.setTitle("Hello World");
15         primaryStage.setScene(new Scene(root, width: 300, height: 275));
16         primaryStage.show();
17     }
18
19
20 public static void main(String[] args) { launch(args); }
21
22
23
24
```

Glavna klasa u aplikaciji mora proširivati `javafx.application.Application` klasu. To omogućava da aplikacija koja se kreira ima sve odlike JavaFX aplikacije.

`void start(Stage primaryStage) throws Exception` – metoda predstavlja glavnu ulaznu tačku JavaFX aplikacija. Poziva se odmah nakon inicijalizacije. Svaka klasa koja proširuje `javafx.application.Application` klasu mora da redefiniše ovu metodu.

`void launch(String... arg0)` – metoda koja inicijalizuje JavaFX aplikaciju.

Kreiranje HelloWorldApp projekta

2) Osnovni elementi JavaFX aplikacije

```
Main.java x Controller.java x sample.fxml x
1 package sample;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.stage.Stage;
8
9 public class Main extends Application {
10
11     @Override
12     public void start(Stage primaryStage) throws Exception{
13         Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));
14         primaryStage.setTitle("Hello World");
15         primaryStage.setScene(new Scene(root, width: 300, height: 275));
16         primaryStage.show();
17     }
18
19     public static void main(String[] args) { launch(args); }
20
21 }
22
23
24
```

JavaFX za opisivanje interfejsa aplikacija koristi Stage-ove i Scene-e.

Stage klasa predstavlja top-level kontejner aplikacije – prozor (naslovna linija sa caption dugmicima). Njegov sadržaj je Scene-a. Mora se učiniti vidljivim da bi bio prikazan.

Scene klasa predstavlja kontejner za sve što se nalazi na Stage-u – sve što se nalazi na prozoru. Sadržaj scene formira hijerarhijsku strukturu grafa, što omogućava da aplikacija bude responsive na promenu veličine prozora. Postavljanjem veličine scene postavlja se veličina prozora. Naša scena sadrži graf root koji je оформljen učitavanjem sample.fxml.

Kreiranje HelloWorldApp projekta

2) Osnovni elementi JavaFX aplikacije

```
Main.java x Controller.java x sample.fxml x
1 <?import javafx.scene.layout.GridPane?>
2
3 <?import javafx.scene.control.Button?>
4 <?import javafx.scene.control.Label?>
5 <GridPane fx:controller="sample.Controller"
6     xmlns:fx="http://javafx.com/fxml" alignment="center" hgap="10" vgap="10">
7     <Label text="Klik me :)"></Label>
8     <Button text="Hello world" onAction="#helloWorld"></Button>
9 </GridPane>
```

`javafx.scene.layout.GridPane` – smešta unutrašnje čvorove u redove i kolone. Definisan je i kontroler koji može da obrađuje događaje, kao i poravnanje i razmaci između kolona i redova.

`javafx.scene.control.Label` ima svoju tekstualnu vrednost

`javafx.scene.control.Button` ima svoju tekstualnu vrednost i definisanu funkciju koja obrađuje klik na dugme

Poziv odgovarajuće funkcije iz Controlera

Kreiranje HelloWorldApp projekta

2) Osnovni elementi JavaFX aplikacije

```
1 package sample;
2
3 import javafx.event.ActionEvent;
4
5 public class Controller {
6
7     public void helloWorld(ActionEvent actionEvent) {
8         System.out.println("Hello world");
9     }
10 }
```

ActionEvent predstavlja događaj koji se odigrao (klik na dugme, pritisnuto dugme na tastaturi...)

void helloWorld(ActionEvent actionEvent) je funkcija koja obradjuje događaj koji se desio

Kreiranje HelloWorldApp projekta

3) Dinamičko dodavanje elemenata

```
@Override
public void start(Stage primaryStage) throws Exception{
    Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));

    GridPane gridPane = (GridPane) root;
    Button btn = new Button("New button");
    btn.setOnAction(event -> {
        System.out.println("New button clicked");
    });

    gridPane.add(btn, columnIndex: 0, rowIndex: 1);

    primaryStage.setTitle("Hello World");
    primaryStage.setScene(new Scene(gridPane, width: 300, height: 275));
    primaryStage.show();
}
```

`void javafx.scene.control.Button
 .setOnAction(EventHandler<ActionEvent>)`

– metoda koja postavlja zadati rukovalac događajem pritiska dugmeta.

Lambda funkcija za reagovanje na događaje od strane korisnika (rukovalac događajem). Poziva se kada se pritisne dugme.

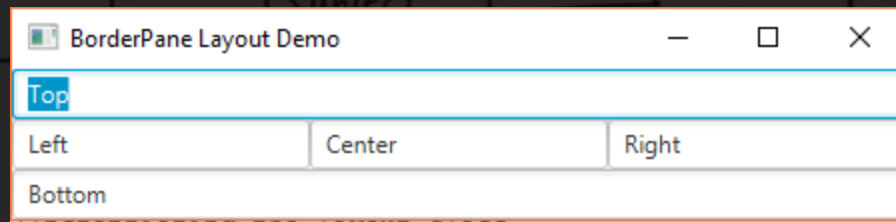
`javafx.scene.layout.GridPane` – JavaFX layout.
`gridPane.add(Node arg0, int column, int row)` – metoda za dodavanje elementa na grid layout u odredjenu kolonu i red.

JavaFX Layout Panes

- Omogućavaju automatsku preraspodelu prostora pri promeni veličine prozora.
- Osnovna klasa iz koje su izvedeni svi layout-i je `javafx.scene.layout.Pane`.
- Moguće je ručno raspoređivanje elemenata po layout-u postavljanjem apsolutne pozicije elementa (tada se koristi `Pane`). Tada nema preraspodele prostora prilikom promene veličine prozora
- Layout Pane-ovi u JavaFX-u:
 - `BorderPane`
 - `HBox`
 - `VBox`
 - `StackPane`
 - `GridPane`
 - `FlowPane`
 - `TilePane`
 - `AnchorPane`

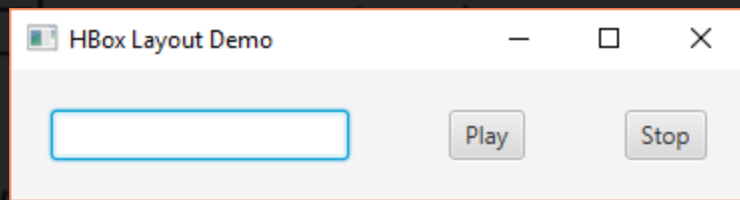
BorderPane

- Deli prozor na pet regiona: top, left, right, bottom i center.
- Nepopunjen prostor pripada centralnom regionu.
- Ukoliko je prozor manji od prostora potrebnog za sve regione, oni se mogu preklapati, pri čemu se onaj region koji je kasnije dodat nalazi iznad regiona koji su dodati pre njega.
- Pogodan je za prozore koji imaju tool bar (u top regionu), status bar (u bottom regionu), panel za navigaciju (u left regionu), dodatne informacije (u right regionu) i prostor za rad (u center regionu).



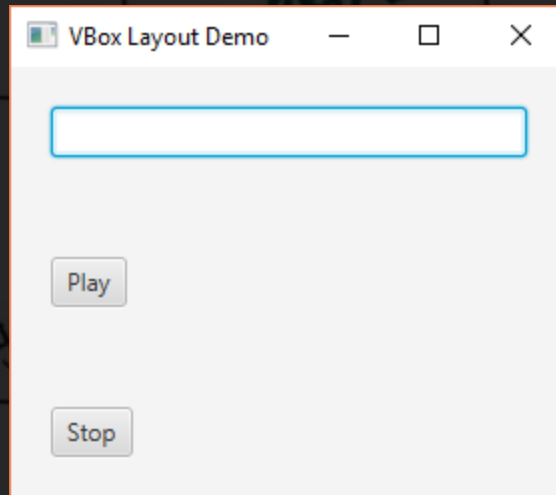
HBox

- Postavlja elemente layout-a u jedan horizontalan red.



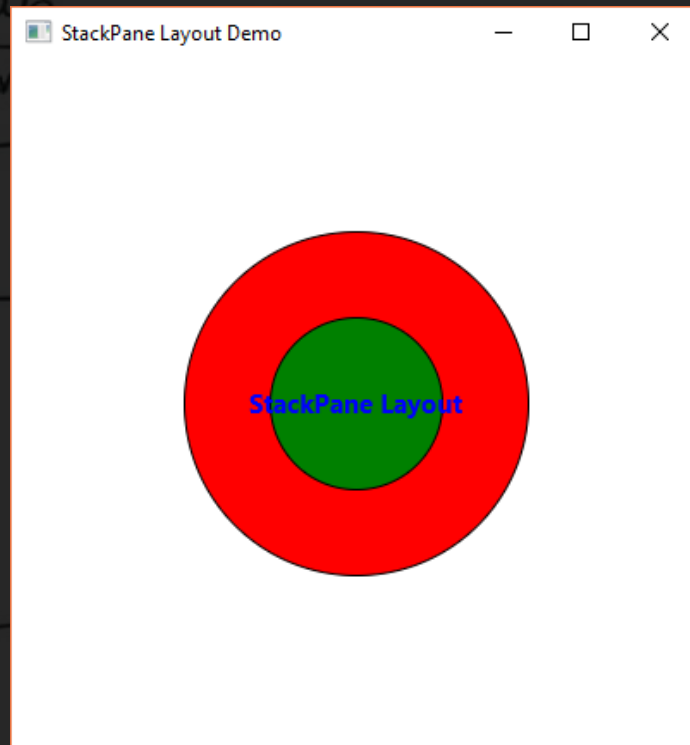
VBox

- Postavlja elemente layout-a u jednu vertikalnu kolonu.



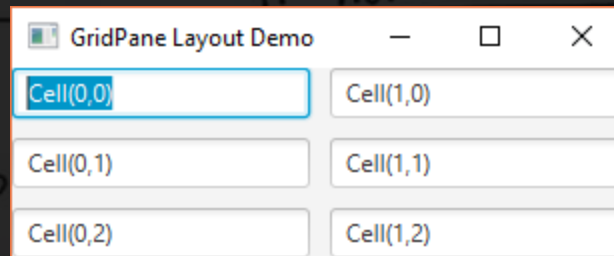
StackPane

- Postavlja elemente layout-a jedan preko drugog, u stek.
- Element koji je prvi dodat nalazi se na dnu steka elemenata.
- Pozicija elementa na ovom layout-u može se podešavati postavljanjem odgovarajućeg alignment-a (utiče na sve elemente layout-a) i margin-a (za svaki element pojedinačno).



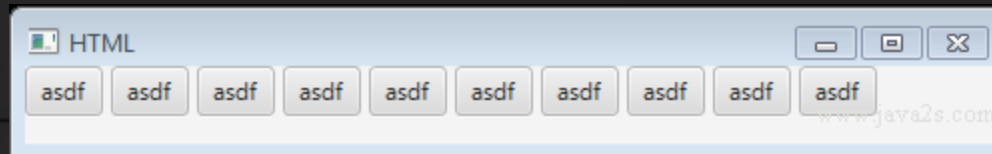
GridPane

- Postavlja elemente layout-a u mrežu redova i kolona.
- Pogodan je za kreiranje formi.



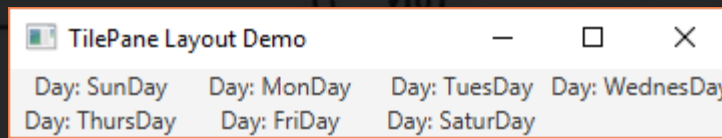
FlowPane

- Obavija sve sadržane elemente – horizontalan (stavlja elemente u redove) i obavija ih po širini elementa, vertikalni (stavlja elemente u kolone) i obavija ih po visini elementa.



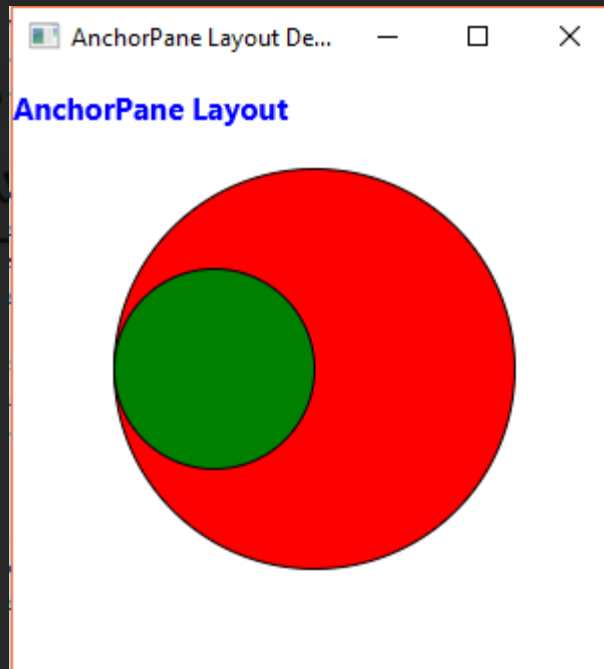
TilePane

- Sličan je FlowPane-u, osim što su svi sadržani elementi uniformne veličine.



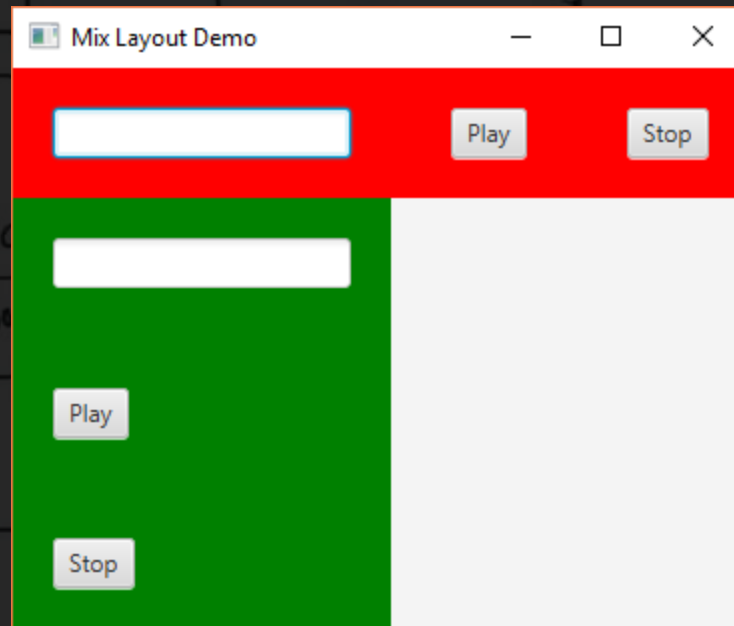
AnchorPane

- Postavlja elemente layout-a na odgovarajuću udaljenost (offset) od njegovih ivica (top, bottom, left i right).
- Prilikom dodavanja elementa ovom layout-u neophodno mu je postaviti bar jedan offset.



Kombinacija Layout-ova

- HBox layout u top regionu BorderLayout layout-a, VBox layout u left regionu BorderLayout layout-a.

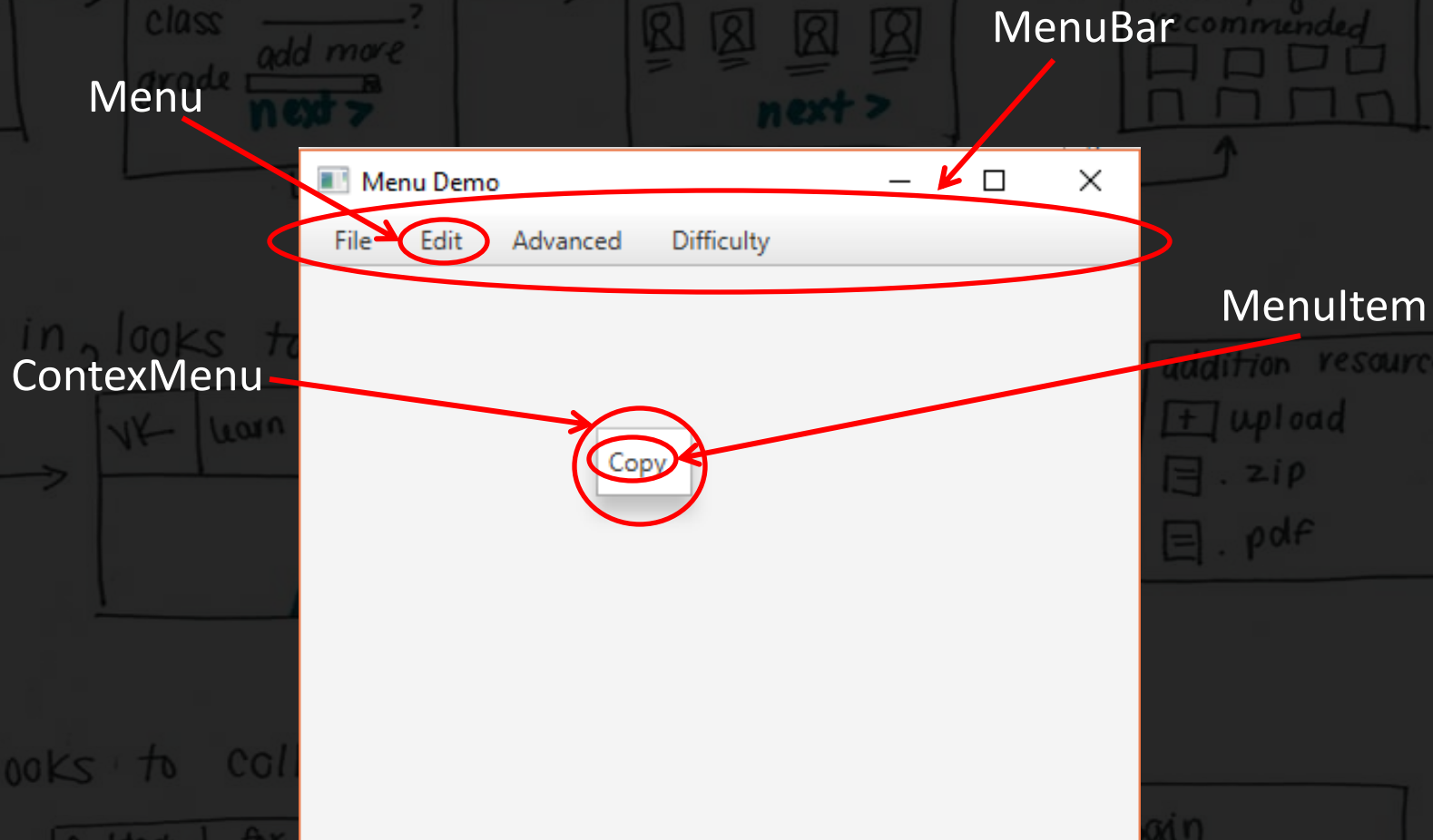


JavaFX Meniji

- JavaFX omogućava kreiranje raznih vrsta menija i njihovih sadržaja:

- MenuBar
- Menu
- MenuItem
- CheckMenuItem
- RadioMenuItem
- SeparatorMenuItem
- ContextMenu

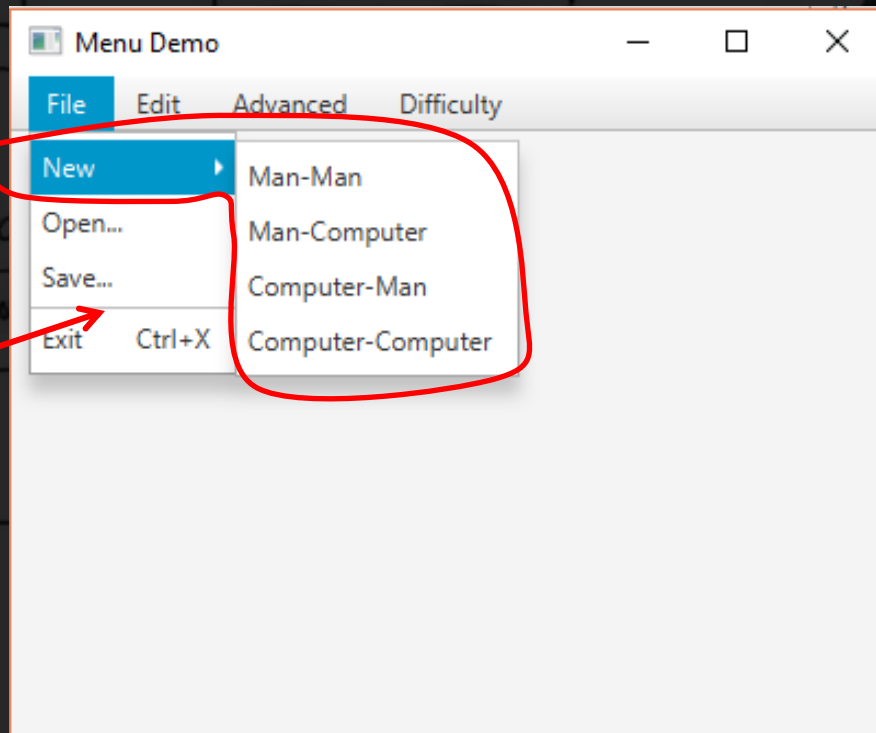
JavaFX Meniji



JavaFX Meniji

Menu (podmeni)

SeparatorMenuItem

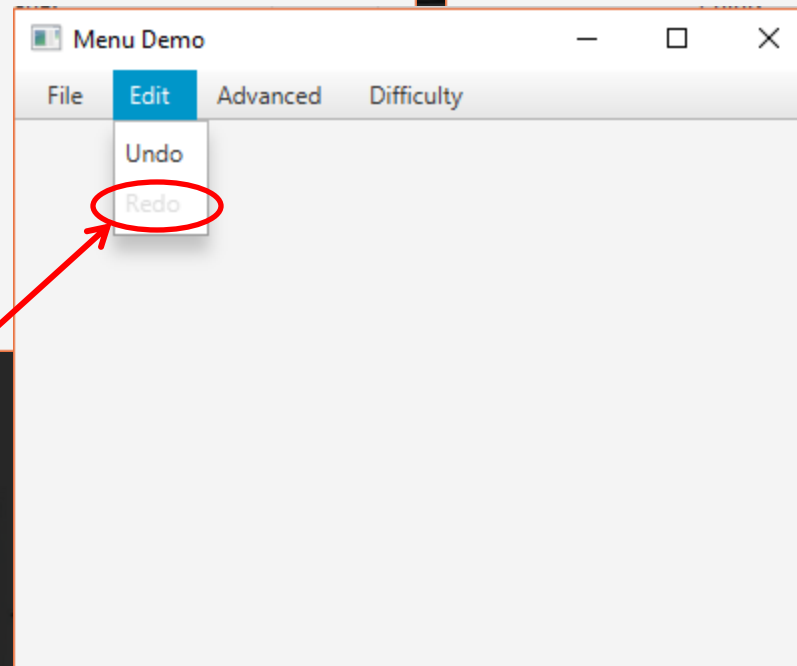
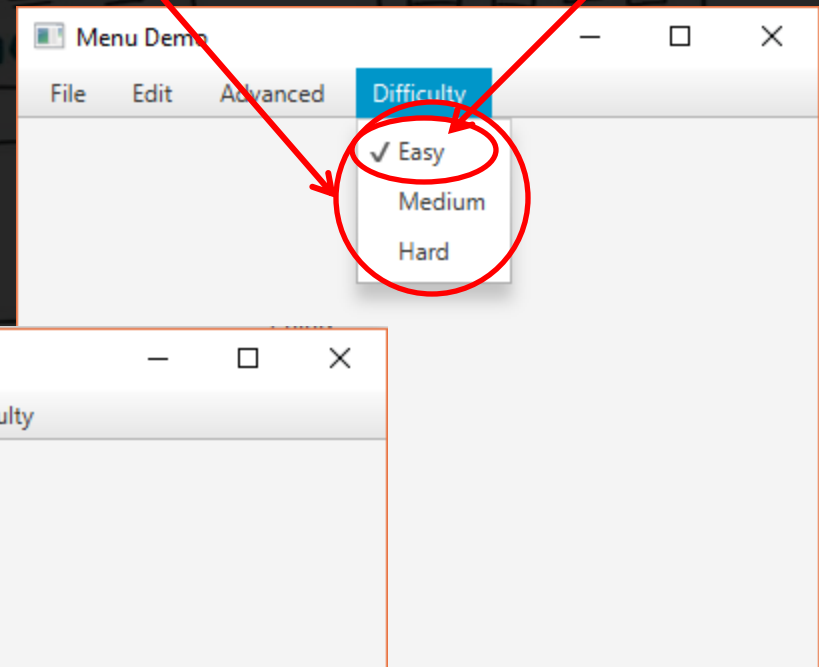
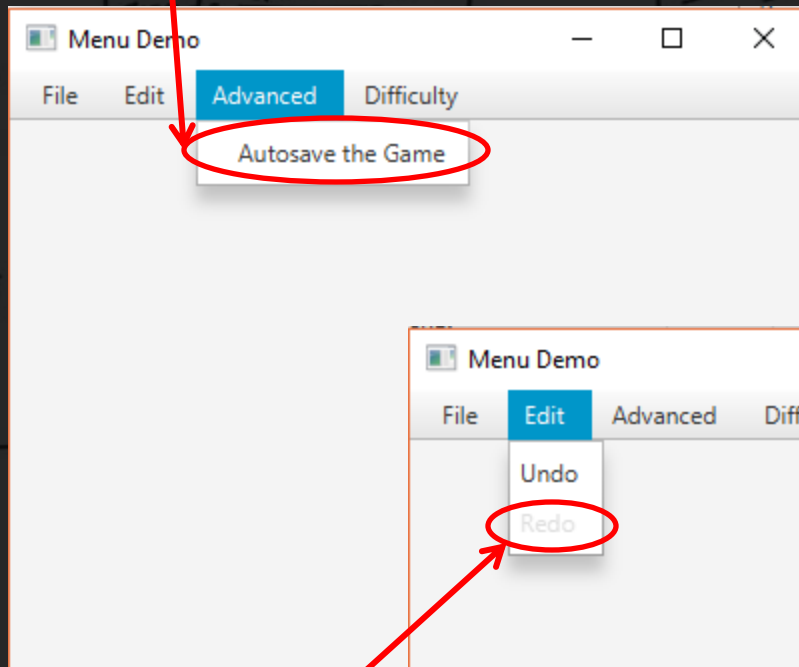


JavaFX Meniji

CheckMenuItem

ToggleGroup

RadioMenuItem



MenuItem
(onemogućeno - disabled)