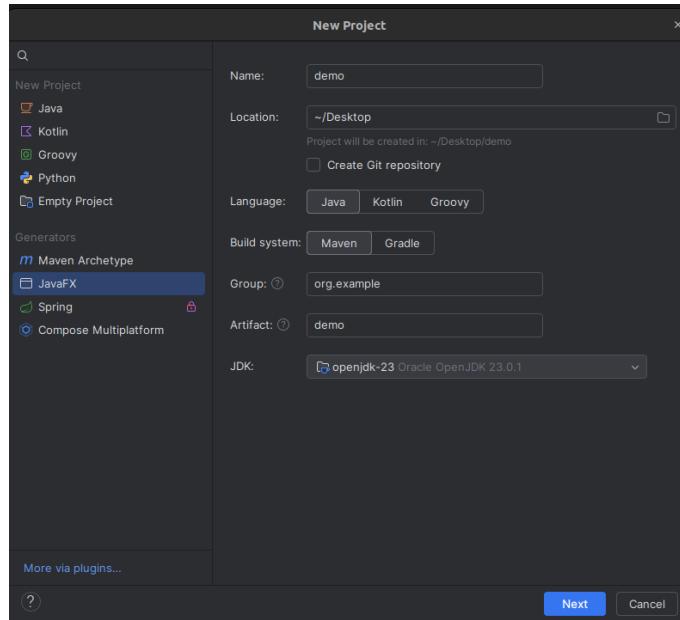
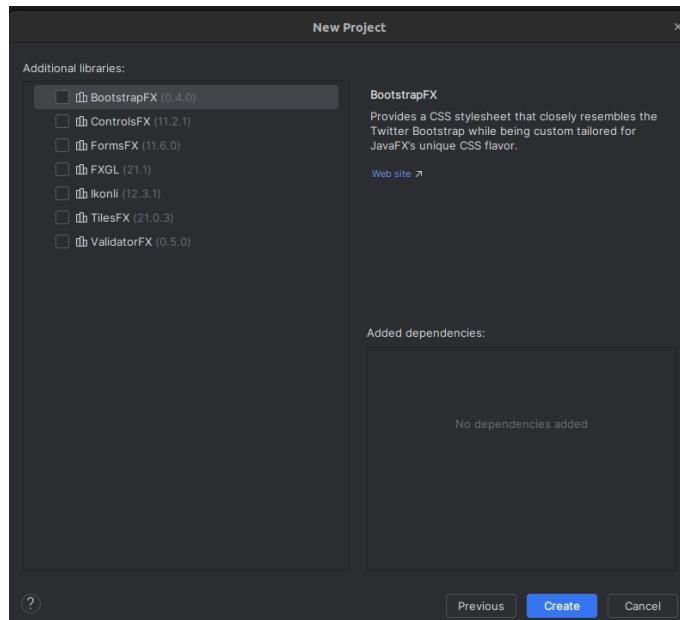


Uputstvo za korišćenje biblioteke JavaFX

Prvi način: Napraviti projekat JavaFX i automatski će se generisati paket u okviru kojeg možete pisati klase koje koriste biblioteku JavaFX (slike 1, 2, 3, 4).



Slika 1: Kreiranje projekta



Slika 2: Za potrebe kursa nije neophodno uključivati dodatne opcije

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** On the left, it shows a project named "demo" with subfolders ".idea", ".mvn", "src" containing ".gitignore", "mvnw", "mvnw.cmd", and "pom.xml".
- Code Editor:** The main window displays the content of `HelloApplication.java`. The code is as follows:

```
1 package org.example.demo;
2
3 import ...
4
5 public class HelloApplication extends Application {
6     @Override
7     public void start(Stage stage) throws IOException {
8         FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));
9         Scene scene = new Scene(fxmlLoader.load(), 320, 240);
10        stage.setTitle("Hello!");
11        stage.setScene(scene);
12        stage.show();
13    }
14
15    public static void main(String[] args) { launch(); }
16}
```

Slika 3: Automatski generisane klase

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** On the left, it shows a project named "demo" with subfolders ".idea", ".mvn", "src" containing "main" (which has "java" and "org.example.demo" subfolders), ".gitignore", "mvnw", "mvnw.cmd", and "pom.xml".
- Code Editor:** The main window displays the content of `HelloApplication.java`. The cursor is positioned over the class name `HelloApplication`.

Slika 4: Pozicioniranje do odgovarajuće klase

Ukoliko želite da imate više paketa u okviru istog projekta čije klase mogu koristiti *JavaFX* grafičke komponente neophodno je izmeniti *module-info.java* datoteku kao na slici 5.

The screenshot shows the IntelliJ IDEA interface. On the left, the 'Project' tool window displays the project structure under 'demo'. It includes '.idea', '.mvn', 'src' (containing 'main' with 'java' and 'zadatak2' (containing 'KlasaGUI'), 'zadatak3' (containing 'KlasaGUI')), 'resources', 'target', '.gitignore', 'mvnw', 'mvnw.cmd', 'pom.xml', 'External Libraries', and 'Scratches and Consoles'. A file named 'module-info.java' is selected in the 'src/main/java/zadatak3' directory. On the right, the code editor shows the content of 'module-info.java':

```
1 module org.example.demo {  
2     requires javafx.controls;  
3     requires javafx.fxml;  
4  
5     opens zadatak2 to javafx.fxml;  
6     opens zadatak3 to javafx.fxml;  
7     exports zadatak2;  
8     exports zadatak3;  
9 }  
10 }
```

Slika 5: Dodavanje paketa koji koriste biblioteku *JavaFX*

Drugi način: Instalirati *JavaFX* biblioteku i napraviti standardni *IntelliJ* projekat. U okviru strukture projekta dodati putanju do biblioteke *JavaFX* i dodati opcije za pokretanje virtuelne mašine. Koraci se mogu vidjeti na [linku](#).

Napomena: Prvi način je popularniji za novije verzije okruženja *IntelliJ* zato što nije potrebno eksplisitno dodavanje putanje do biblioteke i opcija za virtuelnu mašinu.