

Téma

Úvod do formulářových aplikací s GUI s využitím JavaFX.
Vývoj FX aplikací v NetBeans 12.6

Cíl

- Vytvoření jednoduché ukázkové okenní aplikace
- Seznámení se s aplikací Scene Builder
- Prozkoumání struktury aplikace

Instalace JavaFX a SceneBuilder

Instalace nástrojů pro vývoj JavaFX aplikací

JavaFX byla součástí vývojového prostředí NetBeans do verze 8.2. Tato verze je již zastaralá, včetně odpovídající verze Java8. Pro vývoj aplikací je nutno mít k dispozici rovněž Java JDK. Na stránkách <https://netbeans.apache.org/download/archive/index.html> již není oficiálně k dispozici. Je však možno stále stáhnout např. NetBeans 8.2 společně JDK některých archívů, např. https://archive.org/details/jdk-8u111-nb-8_2. Pro vývoj JavaFX aplikací s využitím FXML je možno použít Gluon SceneBuilder verze 8.5.0 (<https://gluonhq.com/products/scene-builder/>).

Od přechodu NetBeans pod Apache (verze 9 a vyšší) již není JavaFX standardní součástí NetBeans a je třeba tuto technologii doinstalovat viz např. <https://openjfx.io/openjfx-docs/>

NetBeans 12.6:

<https://netbeans.apache.org> respektive
<https://netbeans.apache.org/download/index.html> respektive
<https://netbeans.apache.org/download/nb126/nb126.html>

JavaFX

Pro vývoj JavaFX aplikací je třeba stáhnout JavaFX knihovnu a propojit ji s NetBeans a projekty, které vytváříte.

<https://openjfx.io> respektive
<https://gluonhq.com/products/javafx/>

Pro stažení a rozbalení postupujte dle pokynů na <https://openjfx.io/openjfx-docs/>.

Více základních informací o technologii JavaFX naleznete na <https://docs.oracle.com/javase/8/javase-clienttechnologies.htm>

Scene Builder 17.0.0

Proto, abyste mohli vytvářet grafické aplikace systémem drag-drop, je třeba doinstalovat nástroj JavaFX Scene Builder. Pro práci ve spojení s NetBeans verze 12 použijeme Gluon Scene Builder verze 17.

<https://gluonhq.com> respektive
<https://gluonhq.com/products/scene-builder/>

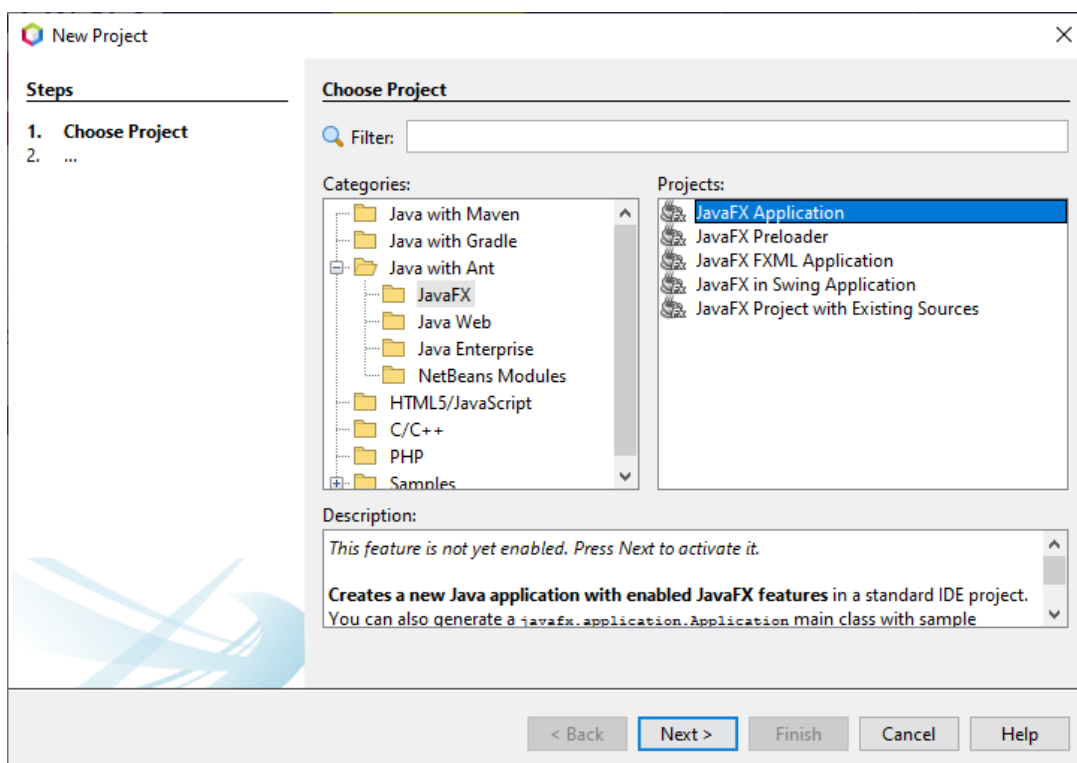
Doporučený studijní materiál pro první JavaFX aplikace

Pro rozšíření znalostí, nebo při nepochopení látky z prvního cvičení, které je naprosto klíčové využijte velmi pěkně zpracovaný materiál na webu: <https://www.itnetwork.cz/java/javafx/java-tutorial-uvod-do-javafx>

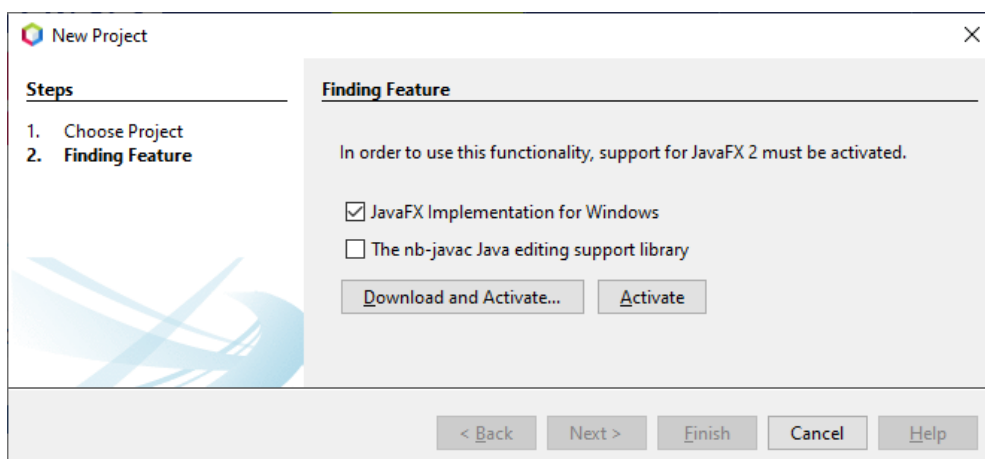
První spuštění NetBeans, doplnění JavaFX,



1. Pokuste se vytvořit nový ANT Java projekt pomocí menu File / New Project / Java with Ant

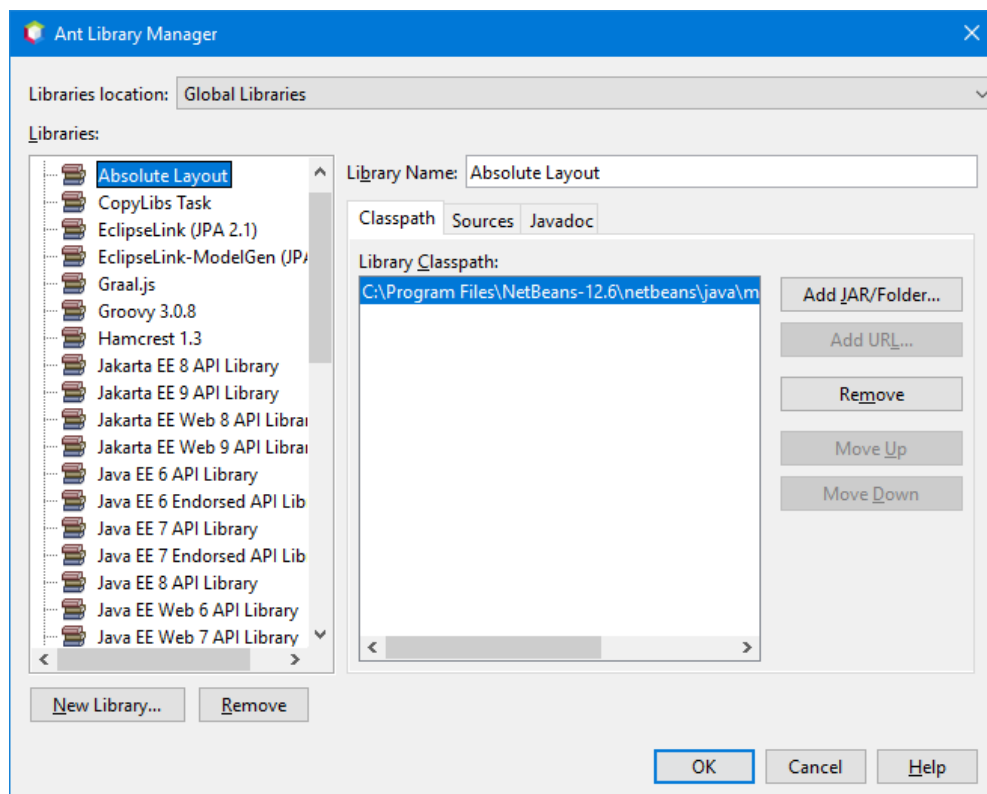


2. Při vytváření prvního projektu (pokud máte NetBeans nově nainstalovány) je třeba stáhnout a aktivovat nb-javac.

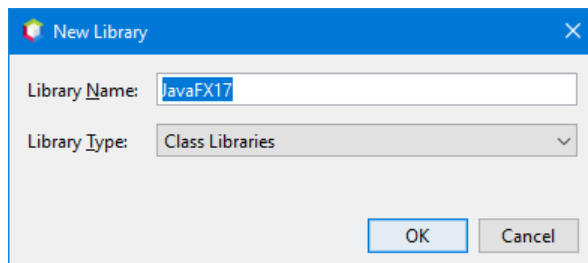


3. Pokud ještě nemáte stáhnutou JavaFX knihovnu, stáhněte ji (<https://gluonhq.com/products/javafx/> - SDK, Win 17.0.2 x64) a rozbalte na vhodné místo, např **D:\JavaFX\javafx-sdk-17.0.2**.

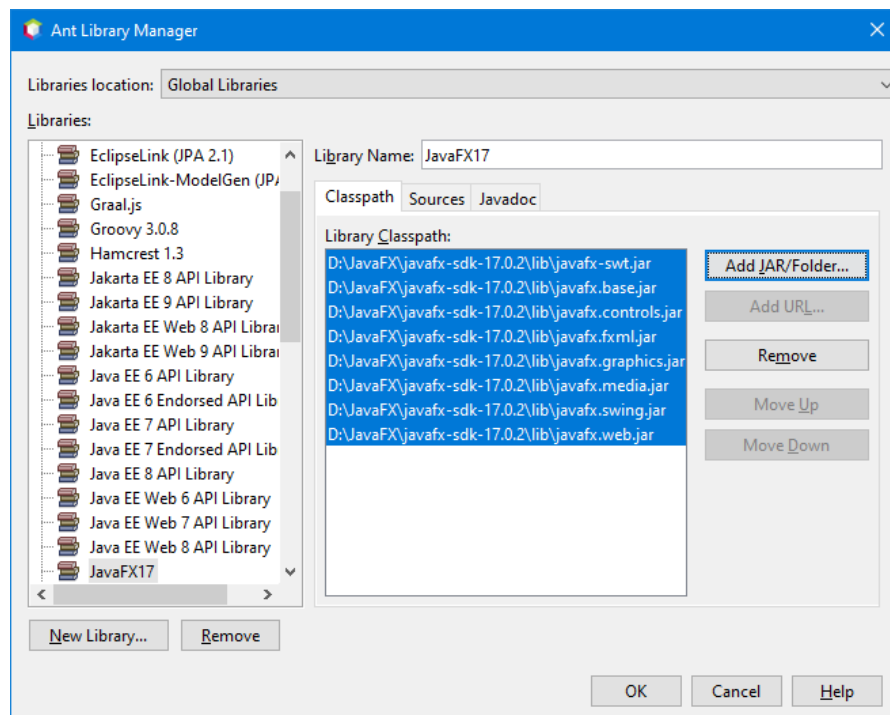
4. V NetBeans přidejte novou knihovnu pomocí menu Tools / Library



Zvolte vhodné pojmenování (např JavaFX17)

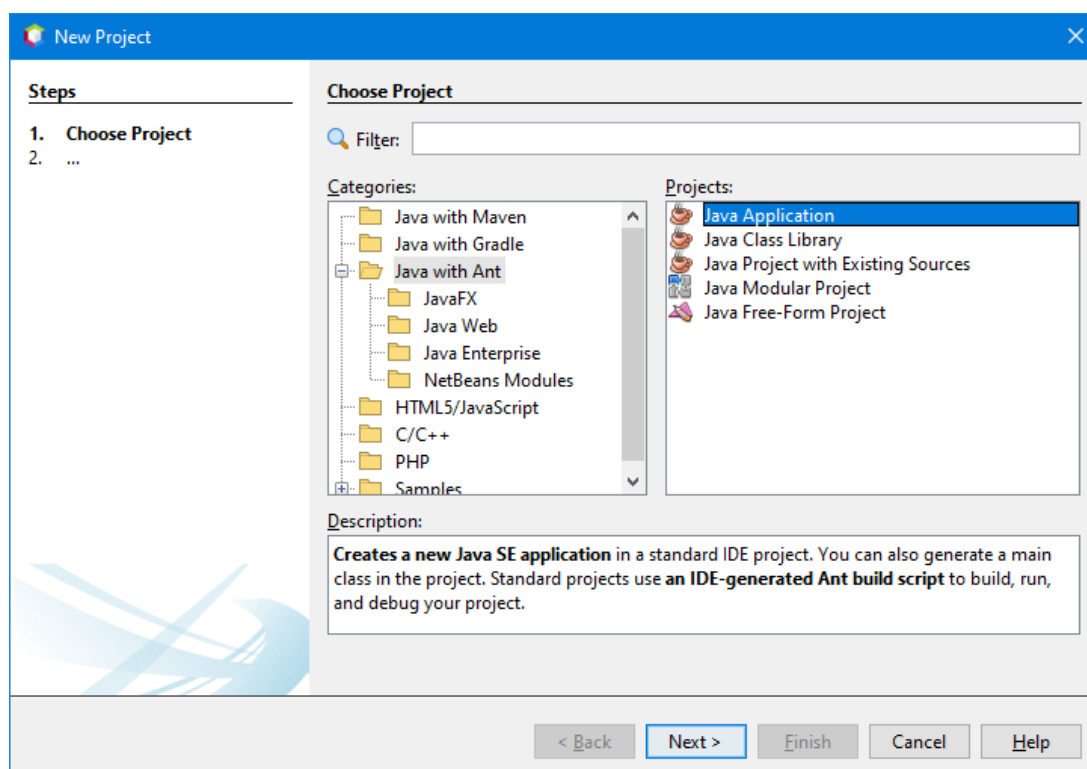


a přidejte všechny jar z adresáře lib ze stažené a rozbalené knihovny. Pozor – NE *.src!

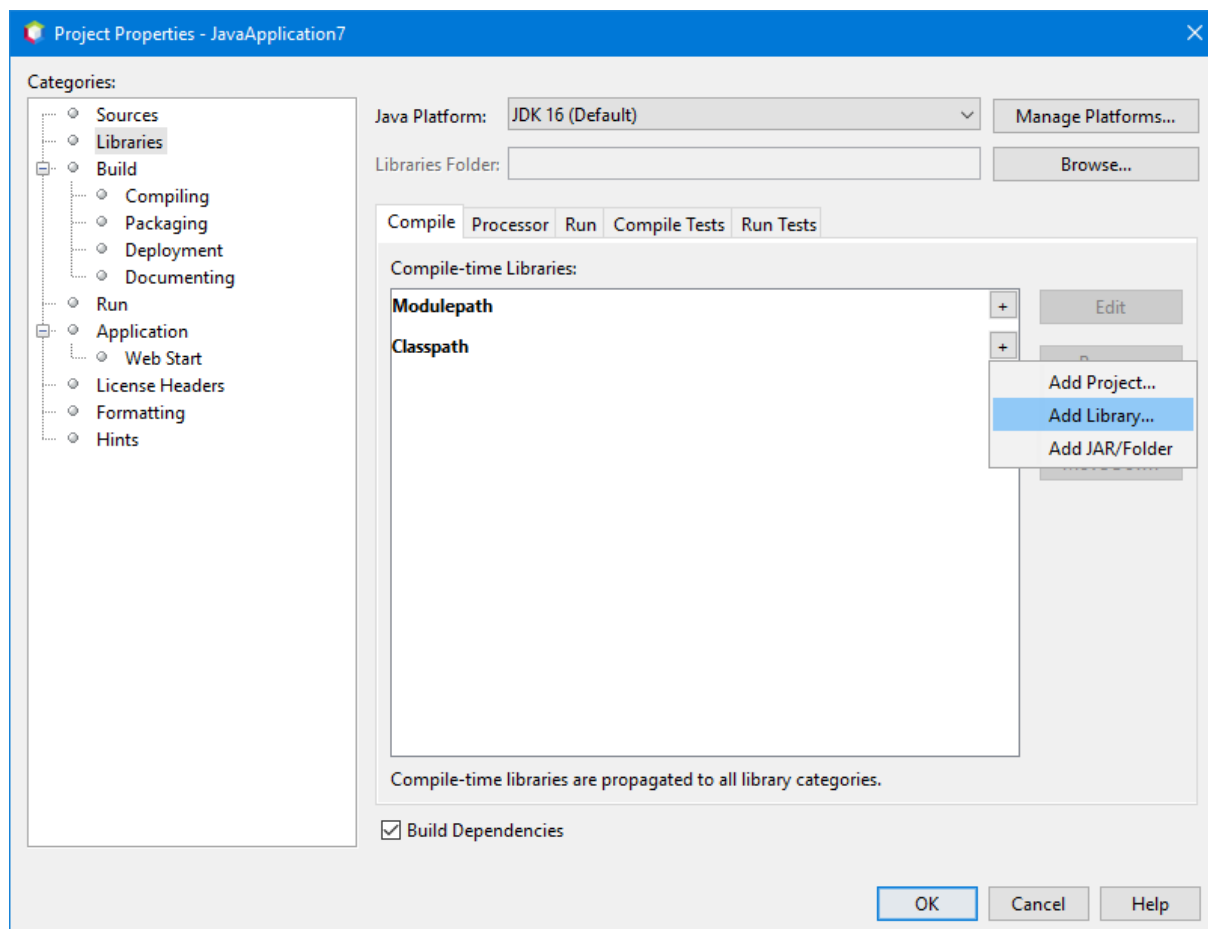


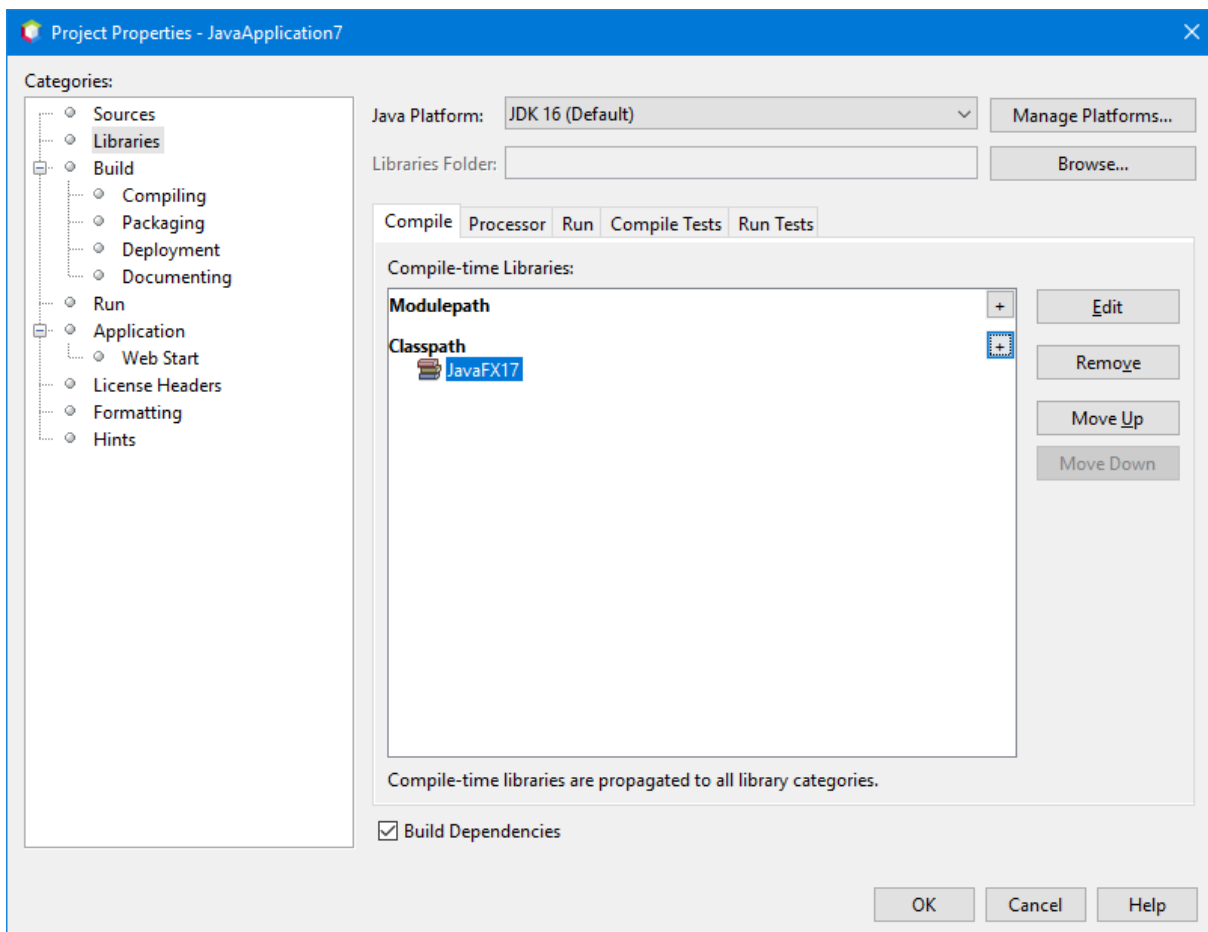
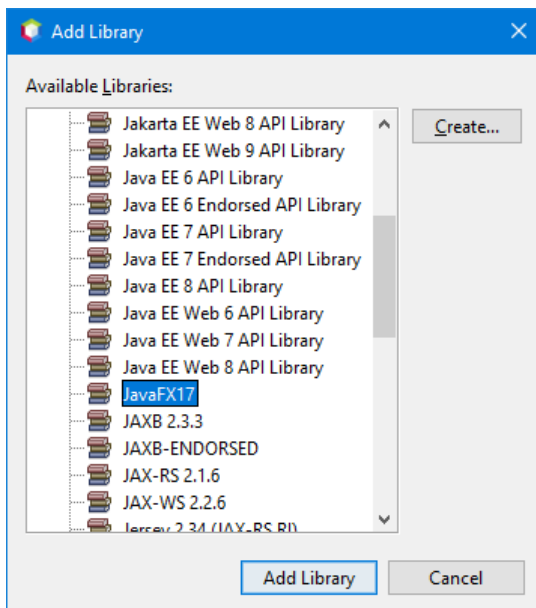
5. Konečně můžeme vytvořit fungující projekt.

Potě pomocí menu File / New Project / Java with Ant / Java Application. Pozor – NE JavaFX!

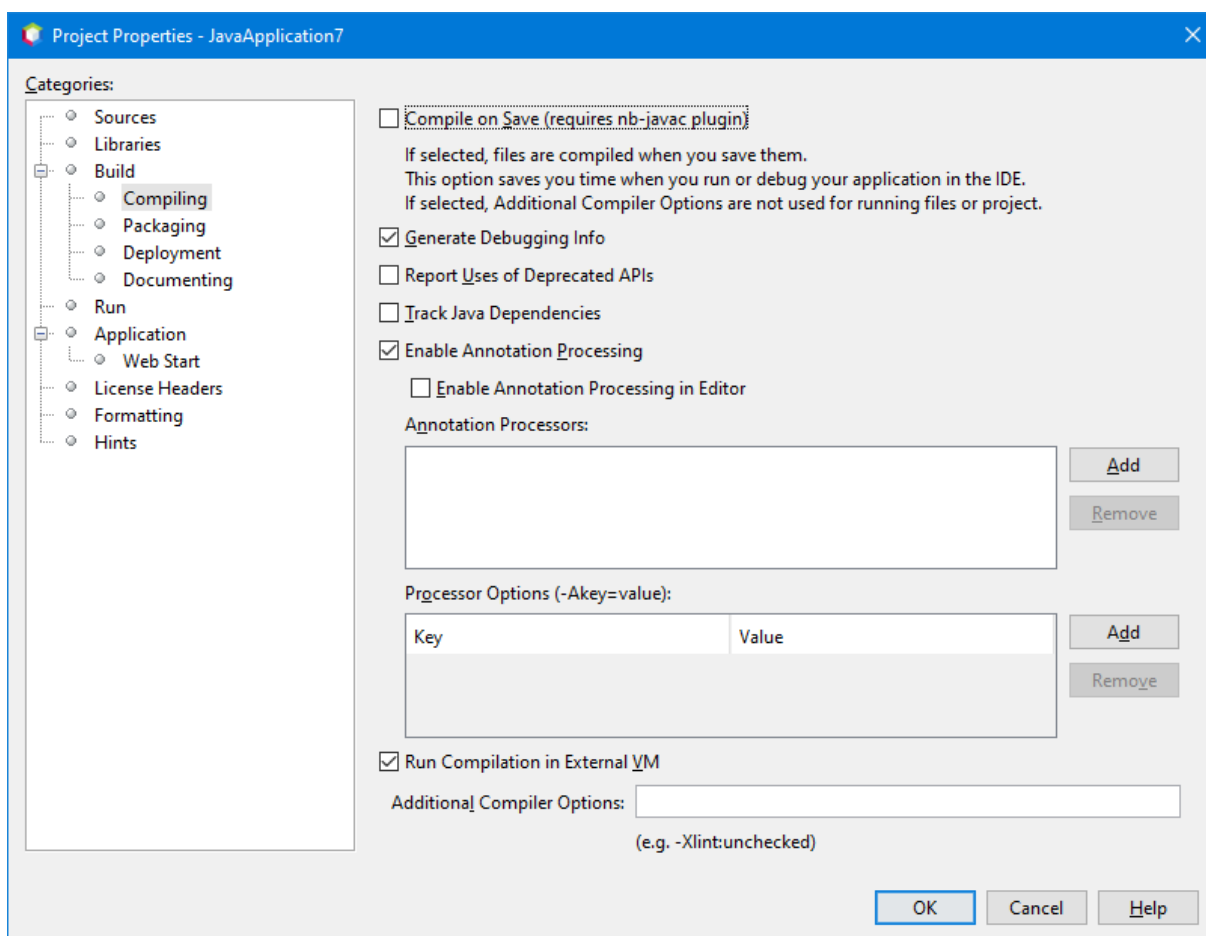


6. Ve vlastnostech projektu (Properties / Libraries / ClassPath+ / Add Library) přidáme do Classpath / Add Library v seznamu již vytvořenou knihovnu JavaFX17.





7. Ve vlastnostech projektu (Properties / Build / Compiling) zkontrolujeme, zda není zaškrtnuto Compile on Save...



Příklad 1: První okenní aplikace bez FXML

Pokusíme se vytvořit okenní aplikaci s jediným tlačítkem na ploše okna, které po stisknutí vypíše (prozatím na konzolu) text „Hello World!“.

Aplikace nebude nevyužívat k definování vzhledu okna a ovládacích prvků samostatný kód v jazyce FXML, vzhled i chování bude definováno v javovském kódu

Jak bylo uvedeno v předchozích krocích, vytvoříme novou Java aplikaci pomocí File/New Project/Java with Ant/Java Application.

- Zvolíme název projektu, a necháme automaticky vytvořit i hlavní a jedinou třídu s metodou main. Vhodně pojmenujeme i balíček.
- Naše nová třída musí být potomkem třídy `javafx.application.Application` (pozor na balíčky, import správné třídy, ALT+ENTER)
- Implementujeme metodu `start`. Ta překrývá metodu ze třídy `Application`. Tato třída je základní třídou pro všechny JavaFx aplikace.
- Doplníme kód metody `start` a volání metody `launch` (hlavní smyčka zpracování událostí v FX aplikaci) v metodě `main`

```
public class cv01_pr01_PrvníAplikace extends Application{

    @Override
    public void start(Stage primaryStage) throws Exception {

        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });

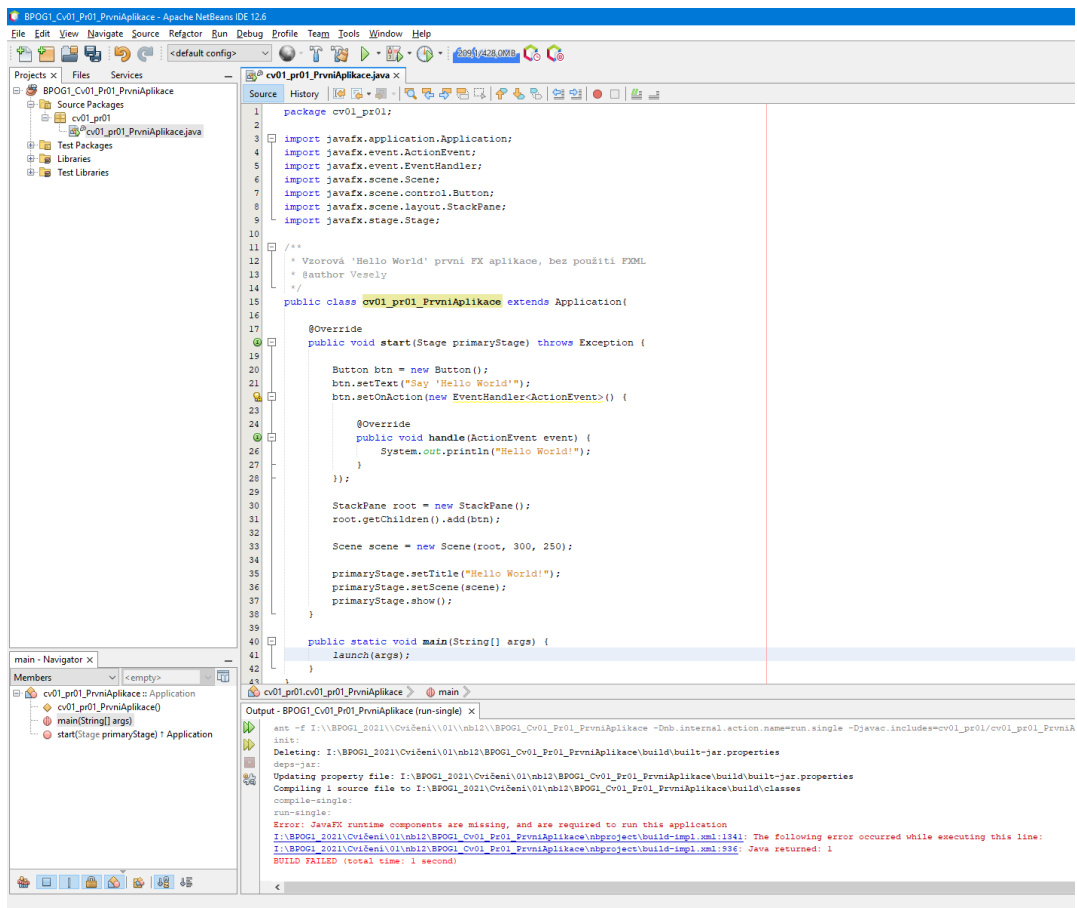
        StackPane root = new StackPane();
        root.getChildren().add(btn);

        Scene scene = new Scene(root, 300, 250);

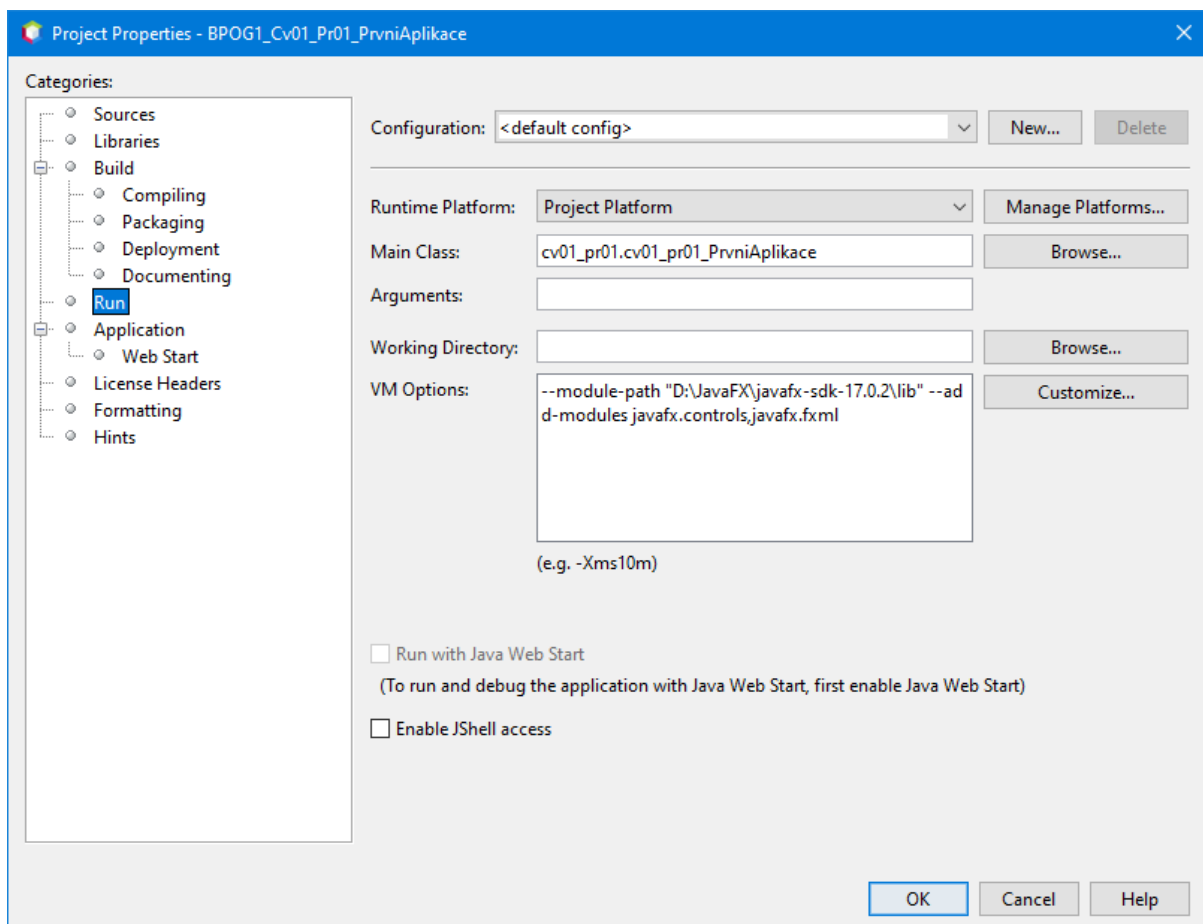
        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

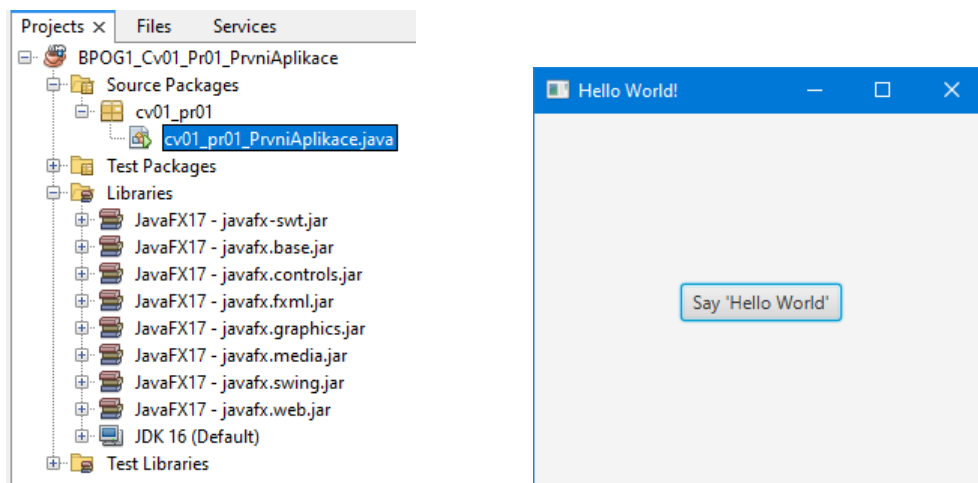
Prohlédněte ukázkový kód a základní strukturu aplikace. Prozkoumejte nastavování vybraných vlastností pro vzhled ovládacích prvků a obsluhu události (stisknutí tlačítka).



Posledním krokem je řešení chyby, vzniklé při pokusu o spuštění. Je třeba Nastavení VM Options. Ve vlastnostech projektu (Properties/Run/VM Options) nastavíme modul dle vlastní cesty ke knihovně JavaFX, např. **--module-path "D:\JavaFX\javafx-sdk-17.0.2\lib" --add-modules javafx.controls,javafx.fxml**



Na následujícím obrázku je struktura projektu a první okenní aplikace po úspěšném spuštění



Příklad 2 – Modifikace předchozího kódu

Vytvořte kopii předchozího 1. příkladu, pomocí refaktoringu přejmenujte balíček (cv01_pr02) a třídu (BPOG1_Cv01_Pr02_ModifikacePrvniAplikace).

Vyzkoušejte nastavení různých vlastností použitých komponent (např. tlačítka).

V rámci samostudia zjistěte a vyzkoušejte další vlastnosti a možnosti nastavení použitých komponent.

Experimentujte s dalšími komponentami.

```
@Override
public void start(Stage primaryStage) {
    Button btn = new Button();
    btn.setText("Say 'Hello World'");

    btn.setRotate(30);

    //btn.setBackground(null);
    btn.setBackground(new Background(
        new BackgroundFill(Color.LIGHTCORAL,
            new CornerRadii(15), new Insets(3))));
    btn.setBorder(new Border(new BorderStroke(Color.BLACK,
        BorderStrokeStyle.SOLID, new CornerRadii(15),
            new BorderWidths(3))));

    btn.setOnAction(new EventHandler<ActionEvent>() {

        @Override
        public void handle(ActionEvent event) {
            System.out.println("Hello World!");
        }
    });

    StackPane root = new StackPane();

    //root.setBackground(Background.EMPTY);
    root.setBackground(new Background(new BackgroundFill(Color.ORANGE,
        new CornerRadii(30), new Insets(30))));

    root.getChildren().add(btn);

    //Scene scene = new Scene(root, 300, 250);
    Scene scene = new Scene(root, 300, 250, Color.SKYBLUE);

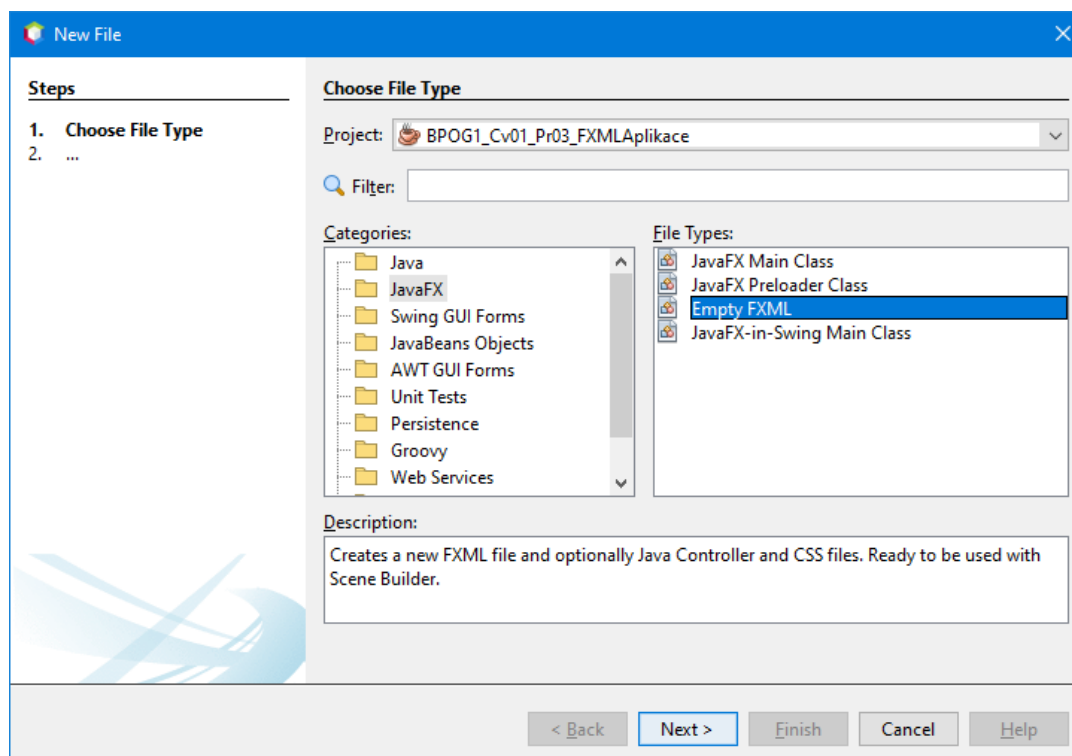
    primaryStage.setTitle("Hello World!");
    primaryStage.setScene(scene);
    primaryStage.show();
}
```

Příklad 3 – První okenní aplikace s využitím aplikace SceneBuilder a jazyka FXML

Pro aplikace JavaFX je typické, že podporují a ze své podstaty velmi usnadňují vytváření vícevrstvé architektury projektu. Typicky MVC. V jednoduchosti to znamená, že je oddělen vizuální vzhled aplikace, její chování (logika) a případně i data.

Aplikace bude mít prakticky stejné chování jako v příkladu 1.

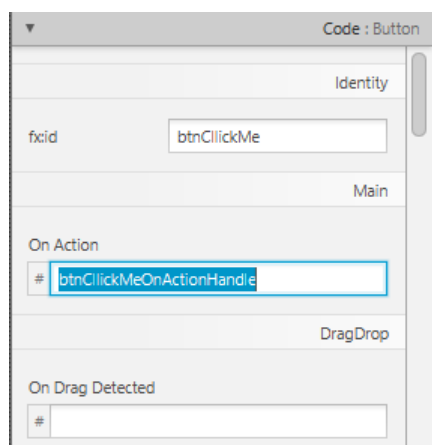
1. Vytvoření nového FXML souboru:

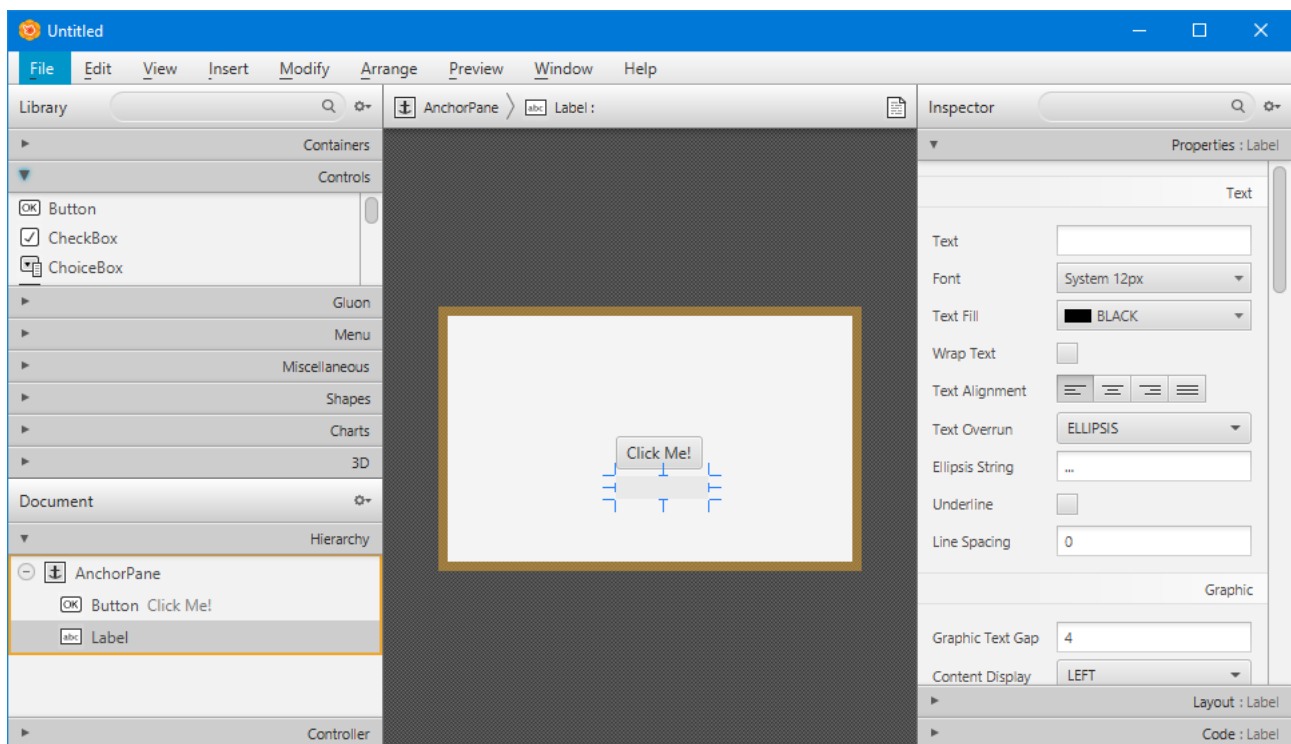


V tomto souboru `FXMLDocument.fxml` bude pomocí jazyka FXML definován vzhled okna a umístění a vzhled dalších ovládacích prvků (tlačítko, textové pole), které toto okno obsahují. Tento soubor lze otevřít v textovém tvaru v prostředí NetBeans (volba Edit v kontextovém menu) nebo v samostatné aplikaci Scene Builder (volba Open v kontextovém menu nebo prostým dvojklikem LT).

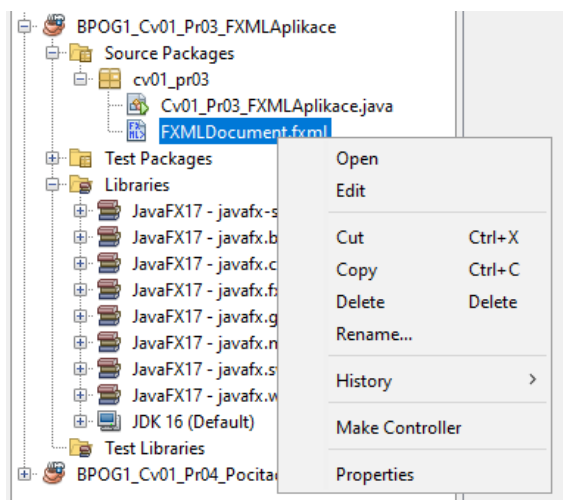
2. FXML soubor upravíme v nástroji SceneBuider.

Prozkoumejte základní možnosti editace v nástroji SceneBuider a jeho části (knihovna komponent, hierarchické zobrazení obsahu, inspektor a částmi pro editaci vlastností, rozložení a kódu), možnost preview, ... Vložte požadovaný kontejner (AnchorPane, Pane, ...) a komponenty tlačítko (Button) a textový popisek (Label). Nastavte umístění a další vlastnosti. Nastavte `fx:id` a pro komponentu tlačítka i odezovou metodu pro standartní událost (OnAction).





3. Vygenerujte k danému FXML dokumentu tzv. controller. Toto generování (přegenerování) je třeba provést při každé (není to úplně pravda, ale je to tak jednodušší) úpravě FXML dokumentu.



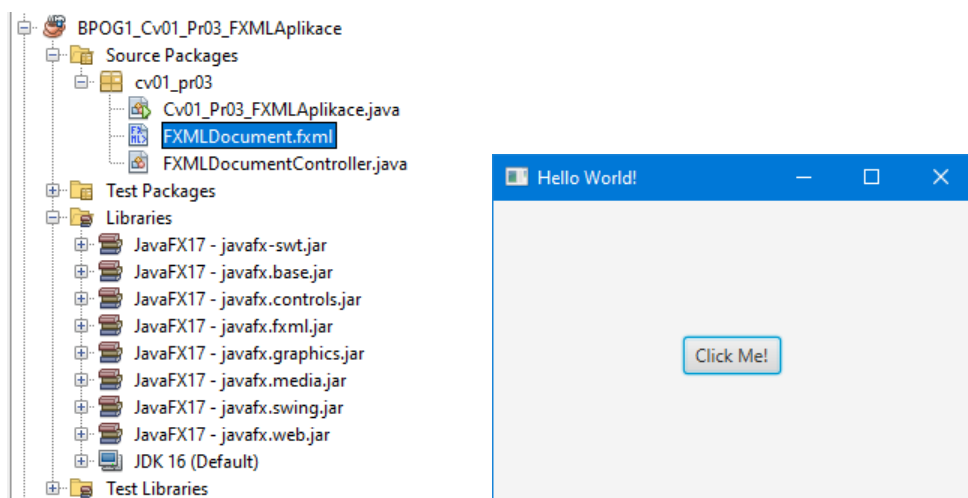
4. Prohlédněte vygenerovaný soubor controlleru a doplňte kód do odezvové metody.

```
public class FXMLDocumentController implements Initializable {
    @FXML
    private Label label;
    @FXML
    private Button btnClickMe;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
    }

    @FXML
    private void btnClickMeOnActionHandle(ActionEvent event) {
        System.out.println("You clicked me!");
        label.setText("Hello World!");
    }
}
```

Výsledná struktura aplikace a vzhled okna spuštěné aplikace jsou na následujících obrázcích.

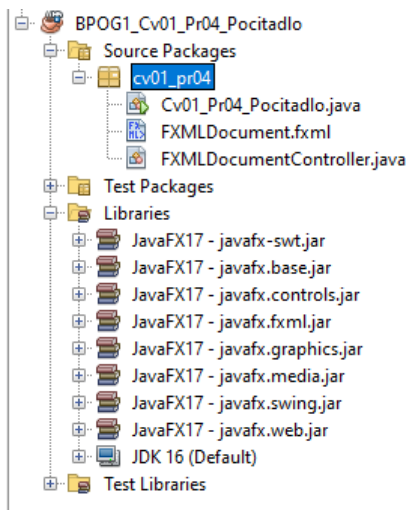


5. Experimentujte s návrhem vzhledu aplikace v programu Scene Builder a sledujte změny v popisu vzhledu v jazyce FXML.

Příklad – Základní počítadlo (čítač)

Vytvořte jednoduchou aplikaci, reprezentující celočíselný čítač. V hlavním okně aplikace je zobrazena aktuální hodnota čítače a dvě tlačítka pro inkrementaci a dekrementaci hodnoty čítače.

Nejdříve je třeba uvědomit si vícevrstvou architekturu, kterou poskytuje JavaFX.



Vydeme ze základní aplikace v příkladu 3, přidáme k ní jedno tlačítko a vytvoříme jednoduchý čítač.

Vzhled budeme upravovat v FXML dokumentu za použití externí aplikace SceneBuilder. Vlastní kód (chování aplikace a ovládacích prvků) budeme vždy vytvářet v příslušném controlleru.

! Před spuštěním JavaFX aplikace je často nutné provést nové sestavení. Pokud vám aplikace nejde, vnut'te spuštění kompilátoru výběrem možnosti „Clean and Build“ v prostředí NetBeans.

Návrh vzhledu okna aplikace může být třeba následující:



Po nastavení vzhledu čítače a nastavení potřebných vlastností fx:id a událostí onAction vygenerujeme nový controller. Do vygenerovaného controlleru (v souboru FXMLDocumentController.java) doplníme odpovídající kód, který bude reprezentovat požadované chování obou tlačítek (+ a -) po jejich stisknutí. Při jejich stisknutí inkrementujeme nebo dekrementujeme hodnotu v proměnné, která datově reprezentuje daný čítač a novou hodnotu čítače zobrazíme v příslušném textovém popisku (label). Nastavení textu v popisku se realizuje pomocí metody `setText()` u objektu příslušné textové komponenty.

Příklad 5 – Rozšíření počítadla

Samostatně proveďte rozšíření aplikace počítadla (čítače) tak, jak vyplývá z následujícího obrázku. Využijte potřebné komponenty (RadioButton, Slider, TextField) z panelu ovládacích prvků (Controls) v programu Scene Builder. Vytvořte odpovídající controller a specifikujte kód, reprezentující chování použitých ovládacích prvků (komponent).

