



Zpracování multimediálních dat – cvičení

Garant předmětu:
doc. Ing. Petr Číka, Ph.D.

Autoři textu:
doc. Ing. Petr Číka, Ph.D.
Ing. David Kohout

BRNO 2023

Autor	doc. Ing. Petr Číka, Ph.D., Ing. David Kohout
Název	Zpracování multimediálních dat – cvičení
Vydavatel	Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací Technická 12, 616 00 Brno
Vydání	první
Rok vydání	2023
Verze	3
Náklad	elektronicky

Tato publikace neprošla redakční ani jazykovou úpravou

Obsah

0	Úvod	3
0.1	Instalace vývojového prostředí knihoven	3
0.2	Základy práce s vývojovým prostředím IntelliJ IDEA	3
0.3	Instalace SceneBuilder	3
1	Cvičení 1 – Úvod do JavaFX	4
1.1	Vytvoření projektu	4
1.2	Tvorba rozhraní	5
1.3	Spuštění hlavního okna	6
2	Cvičení 2 – Převody barevných modelů	7
2.1	Teoretická část	8
2.1.1	Barevný model <i>RGB</i>	8
2.1.2	Barevný model <i>YCbCr</i>	9
2.2	Praktická část	10
2.2.1	Načtení a zobrazení digitálního obrazu	10
2.2.2	Metoda pro získání matic <i>RGB</i> z načteného obrazu	11
2.2.3	Metoda pro vytvoření nového obrazu z matic <i>RGB</i>	12
2.2.4	Zobrazení pouze jedné barevné složky	12
2.2.5	Ověření funkčnosti implementovaných metod	13
2.3	Samostatná práce	13
2.3.1	Na co si dát pozor?	14
2.4	Jak otestovat?	14
2.5	Bonus	15
2.6	Jak odevzdat?	15
3	Cvičení 3 – Vzorkování	16
3.1	Cíle cvičení	16
3.2	Teoretická část	16
3.3	Praktická realizace	18
3.3.1	Zobrazení komponent barevného modelu <i>RGB</i>	18
3.3.2	Podvzorkování modelu <i>YCbCr</i>	19
3.3.3	Nadvzorkování modelu <i>YCbCr</i>	22
4	Cvičení 4 – Objektivní hodnocení kvality	24
4.1	Cíle cvičení	24
4.2	Teoretická část	24
4.2.1	Metody pro měření kvality obrazových dat	24
4.2.2	Subjektivní měření kvality	24
4.2.3	Objektivní měření kvality	25
4.3	Praktická realizace	28
4.3.1	Implementace algoritmů kvality	28
4.3.2	Úprava grafického rozhraní	28

5	Cvičení 5 – Transformace	31
5.1	Cíle cvičení	31
5.2	Teoretická část	31
5.2.1	Transformační kódování	31
5.3	Praktická realizace	37
5.3.1	Vytvoření třídy a kostry metod	37
5.3.2	Implementace metod pro vytvoření transformačních matic libovol- ných rozměrů	37
5.3.3	Implementace obecné metody pro 2D transformaci	38
5.3.4	Propojení se zbytkem aplikace	38
5.3.5	Provedení transformace rozdělením obrazu na bloky velikosti $N \times N$	38
6	Cvičení 6 – Kvantizace	40
6.1	Cíle cvičení	40
6.2	Teoretický úvod	40
6.2.1	Kvantizace	40
6.3	Praktická realizace	43
6.3.1	Vytvoření třídy a kostry metod	43
6.3.2	Implementace metod pro úpravu kvantizačních matic libovolných rozměrů	44
6.3.3	Implementace úprava kvality	44
6.3.4	Implementace metody pro kvantizaci	44
6.3.5	Implementace metody pro inverzní kvantizaci	44
6.3.6	Provedení kvantizace na blocích velikosti $N \times N$	45
6.3.7	Propojení se zbytkem aplikace	45

0 Úvod

Učební text slouží k podpoře cvičení předmětu Zpracování multimediálních dat. Průběh cvičení je rozdělen do dvou částí:

1. V první části semestru bude úvodní cvičení, kde se vytvoří rozhraní aplikace pomocí JavaFX. Následně bude následovat 5 cvičení zaměřených na implementaci algoritmů využívaných při kompresi obrazu a videa technikami JPEG, MPEG, H.26x.
2. Celý semestr - práce na individuálním projektu, od šestého cvičení budou probíhat pouze konzultace k řešení projektů.

0.1 Instalace vývojového prostředí knihoven

1. Nainstalujte Java SE Development Kit v aktuální verzi dostupné [na webových stránkách www.oracle.com](http://www.oracle.com).
2. Nainstalujte aktuální verzi IntelliJ IDEA Community dostupnou [na stránkách www.jetbrains.com](http://www.jetbrains.com).

0.2 Základy práce s vývojovým prostředím IntelliJ IDEA

- Prostudujte základní rozložení a ovládání vývojového prostředí na <https://www.jetbrains.com/help/idea/>
- Naučte se, jak vytvořit aplikaci JAVA - kapitola Create your first application. Naučte se vytvořit spustitelný JAR soubor.
- Naučte se používat klávesové zkratky - Keyboard shortcuts, což Vám velmi pomůže při dalším vývoji aplikací.
- V nastavení (CTRL+ALT+S) je vhodné vyhledat **Code completion** a zde zrušit zaškrtnutí políčka **Match case**.

0.3 Instalace SceneBuilder

Pro vytvoření rozhraní aplikace budeme používat framework JavaFX. Pro usnadnění tvorby rozhraní slouží aplikace SceneBuilder. Tu stáhneme ze stránek <https://gluonhq.com>. Po nainstalování je možné SceneBuilder nalinkovat přímo do IntelliJ, pak je možné rozhraní vytvářet přímo uvnitř IntelliJ, nebo lze jednoduše SceneBuilder spustit. V IntelliJ můžete nainstalovat plugin JavaFX (měl by ale být součástí) a SceneBuilder nalinkovat v nastavení a vyhledat JavaFX (záložka Languages & Frameworks -> JavaFX), zde se vloží cesta k SceneBuilder.exe. Takto je možné otevřít FXML soubory kliknutím pravým tlačítkem a zvolit položku Open In SceneBuilder (2. od konce). Alternativně lze otevřít FXML soubor přímo a v dolní části okna přepnout zobrazení z Text na Scene Builder. Poslední možností je SceneBuilder spustit manuálně a potřebný FXML soubor vyhledat a otevřít z průzkumníku.

1 Cvičení 1 – Úvod do JavaFX

1.1 Vytvoření projektu

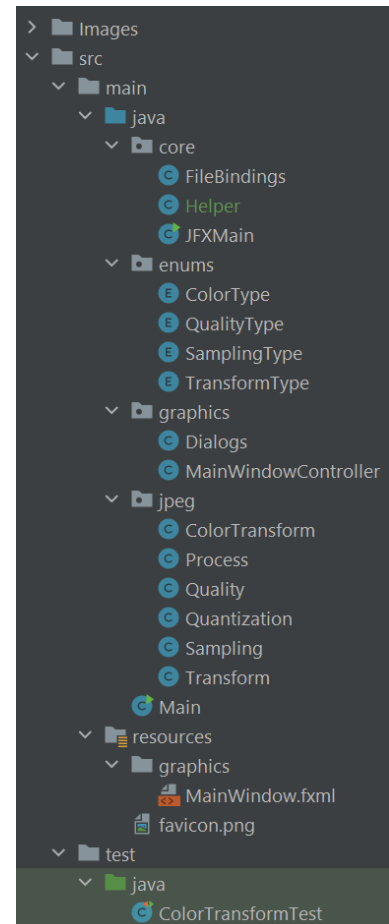
V IntelliJ IDEA si založte nový projekt. Doporučený postup: **File -> New -> Project**. V novém okně stačí vybrat nejzákladnější možnost, tedy **New Project** a zde si nastavit název, umístění, verzi Javy (**Java 11** je v učebně). Poslední je **Build system** ten nastavte na **Maven**.

Projekt je vhodné rozdělit do několika logických celků (balíčků). Ve složce java (složka s kódem) je vhodné použití následujících balíčků (New -> Package): Core, Enums, Graphics a Jpeg. Ve složce resources je dále potřebná složka se stejným názvem, kde se bude nacházet soubor s rozhraním, tedy složka Graphics. Celá struktura projektu, včetně základních tříd je na obrázku 1.1. Tato struktura odpovídá vzorovému programu po všech cvičeních.

Od Javy 9 již není JavaFX součástí oficiálních balíčků Javy, proto je nutné si JavaFX stáhnout jako dodatečnou knihovnu. K tomu nám slouží právě Maven, který jsme si zvolili při tvorbě projektu. Nyní je nutné rozšířit soubor **pom.xml** o pár řádků. Do elementu `<project>` před jeho uzavření vložíme nový element `<dependencies>` a dovnitř vložíme potřebné knihovny:

```
<dependency>
  <groupId>org.openjfx</groupId>
  <artifactId>javafx-controls</artifactId>
  <version>16</version>
</dependency>

<dependency>
  <groupId>org.openjfx</groupId>
  <artifactId>javafx-fxml</artifactId>
  <version>16</version>
</dependency>
```

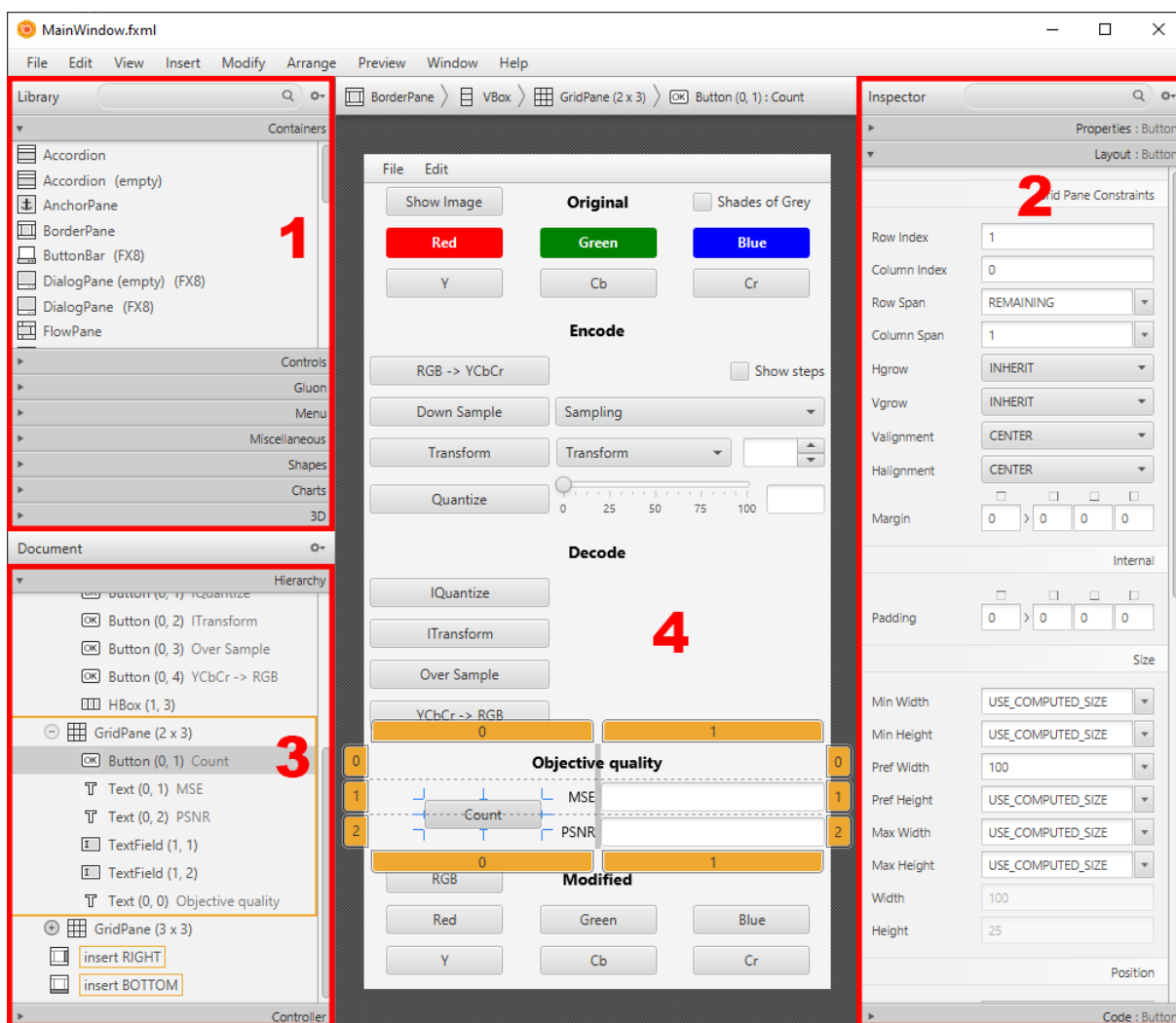


Obrázek 1.1: Struktura

1.2 Tvorba rozhraní

Tvorba rozhraní přes SceneBuilder je velice jednoduchá.

Obrázek 1.2 představuje ovládací prvky aplikace. V prohlížeči komponentů {1} je možné nalézt potřebný ovládací prvek, který se následně přesune do pracovní oblasti, kde budujeme celé rozhraní aplikace {4}. Každý vložený prvek lze upravit v pravém podokně {2}. Tato sekce je rozdělena do 3 částí. První (Properties) upravuje základní nastavení, druhá (Layout) nastavení spojené s umístěním a rozložením v okně. Poslední sekce (Code) řeší propojení s kódem, nejdůležitější položky v této sekci jsou `fx:id` tím můžeme propojit prvek z proměnou v kódu a druhou položkou je `On Action`, která slouží pro nastavení názvu metody, která se provede po např. stisknutí tlačítka.



Obrázek 1.2: Scene Builder

Část {3} představuje hierarchické zobrazení celého rozhraní. Je zde možné vybírat i elementy, které nejsou viditelné v grafickém znázornění. Také je zde možné upravit pořadí prvků (pozn: pořadí prvků souvisí například s ptoklikáním se po elementech tlačítkem tab při spuštění aplikace.) Taktéž se v této části nastavuje Controller, který bude dané okno používat jako propojení s kódem.

Vytvořené rozhraní je možné zobrazit klávesovou zkratkou CTRL+P, případně přes menu Preview -> Show Preview in Window. Poslední, co stojí za zmínku je automatické vytvoření kostry kontrolní třídy, které je možné provést přes View -> Show Sample Controller Skeleton. **(Celé grafické rozhraní je vhodné vytvořit celé najednou a připravit veškeré ID a volání metod. Až je celé okno připravené, teprve poté je vhodné si vygenerovat Controller a vložit jej do kódu.)**

1.3 Spuštění hlavního okna

Pro spuštění aplikace je nutné vytvořit třídu, která provede inicializaci JavaFX, načte vytvořené rozhraní a zobrazí jej do okna. Výsledná třída je znázorněná na obrázku 1.3. Tato třída rozšiřuje třídu Application. Kód uvnitř metody start je v podstatě minimální potřebný kód pro spuštění aplikace. Není nutné nastavovat titulek, ikonu a akci při ukončení. **Aplikace musí být spuštěna metodou main, která obsahuje volání launch. Bohužel od Javy 9 nejde tento main spustit napřímo, a tak je nutné vytvořit další spustitelnou třídu, kde se v metodě main zavolá main této třídy: JFXMain.main(args);.**

```
public class JFXMain extends Application { 2 usages
    private static Stage primaryStage; 6 usages
    public static Scene mainScene; 2 usages

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

        FXMLLoader fl = new FXMLLoader(FilePaths.GUIMain);
        Parent root = fl.load();
        mainScene = new Scene(root);
        primaryStage.setScene(mainScene);

        primaryStage.setTitle("JPEG: Prijmeni VUTID"); //Titulek okna, nastavte svoje
        primaryStage.getIcons().add(FilePaths.favicon); //Přidání ikony aplikace
        primaryStage.show(); //Zobrazí rozhraní

        //Není nutné, umožňuje provést kód po stisku X (př. potvrzení ukončení)
        primaryStage.setOnCloseRequest((e) -> {
            Platform.exit();
            System.exit(status: 0);
        });
    }

    //Nutné mít launch uvnitř main metody.
    //Bohužel nejde spustit přímo z této třídy.
    public static void main(String[] args) { 1 usage
        launch(args);
    }
}
```

Obrázek 1.3: JFXMain