

**Technická univerzita v Košiciach  
Fakulta elektrotechniky a informatiky**

# **Program pre podporu zvukovej audiometrie**

**Príloha D: Systémová príručka**

**Bakalárska práca**

**2024**

**Zora Mária Marcinová**

# Obsah

---

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Prehľad programu . . . . .	1
1.2	Zdrojové súbory . . . . .	1
<b>2</b>	<b>AudiometryApp.java a Main.java</b>	<b>3</b>
<b>3</b>	<b>Definovanie úrovni hlasitosti</b>	<b>4</b>
<b>4</b>	<b>Testovanie</b>	<b>5</b>
4.1	Zvukové súbory . . . . .	5
4.2	Trieda Sound.java . . . . .	6
4.3	Trieda Playlist.java . . . . .	6
4.4	Trieda UserEntry.java . . . . .	6
4.5	Trieda ResultEntry.java . . . . .	6
4.6	Trieda Test.java . . . . .	6
4.6.1	setVolume() . . . . .	7
4.6.2	initializeSoundList() . . . . .	7
4.6.3	playSound() . . . . .	7
4.6.4	playAll() . . . . .	7
4.6.5	markClick() . . . . .	7
4.6.6	recordResults() . . . . .	7
<b>5</b>	<b>Používateľské rozhranie</b>	<b>8</b>
<b>6</b>	<b>Kompilácia projektu</b>	<b>10</b>

# Zoznam obrázkov

---

3.1	Obsah súboru zvukove_urovne.config. . . . .	4
4.1	UML diagram systému . . . . .	5
5.1	Diagram postupnosti obrazoviek implementovaného riešenia . . .	9

# Zoznam tabuliek

---

5.1	Obrazovky rozhrania . . . . .	8
-----	-------------------------------	---

# 1 Úvod

---

## 1.1 Prehľad programu

Program testuje sluch používateľa prehratím 9 zvukových stôp. Tieto zvuky sa prehrávajú na určených úrovniach systémovej hlasitosti zariadenia, ktoré sa dajú vopred nastaviť v konfiguračnom súbore. Každý zvuk sa najprv prehrá na najnižšej úrovni hlasitosti. Ak používateľ na daný zvuk nereaguje, prehrá sa na vyššej úrovni.

Výsledky sú uložené v jednej tabuľke. Táto tabuľka zaznamenáva poradie prehrávaných zvukov a reakcie používateľa.

Aplikácia bola vyvinutá v programovacom jazyku Java a používateľské rozhranie bolo vytvorené pomocou knižnice JavaFX.

## 1.2 Zdrojové súbory

Komponenty aplikácie sú rozdelené do dvoch programových sekcií.

Testovacia sekcia pozostáva z:

- 5 Java tried,
- 1 enumeračný typ,
- súbor nircmd.exe,
- dva priečinky so zvukovými nahrávkami WAV.

Sekcia používateľského rozhrania obsahuje:

- 6 súborov FXML,
- 7 tried Java,
- súbor ikon,
- 1 súbor CSS.

Zdrojový kód aplikácie navyše obsahuje 2 Java triedy obsahujúce hlavnú funkciu programu a konfiguračný súbor na definovanie úrovni hlasitosti na testovanie.

## 2 AudiometryApp.java a Main.java

---

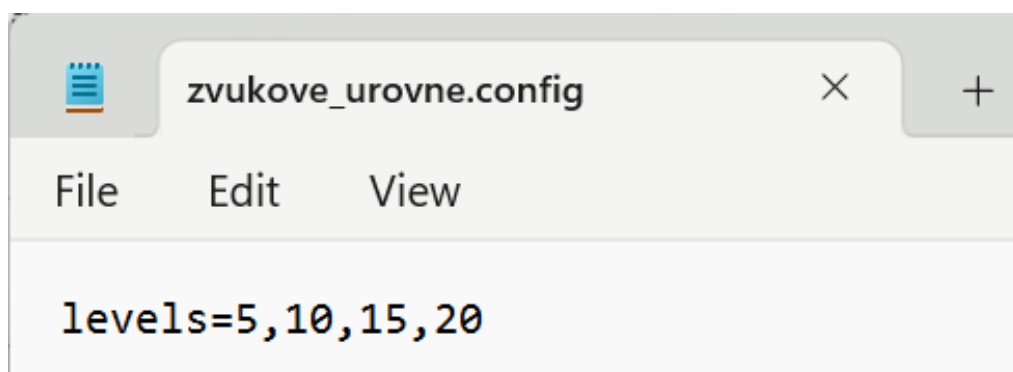
Trieda *AudiometryApp.java* dedí od triedy *Application.java* z knižnice JavaFX. Je potrebná na spustenie a spravovanie JavaFX aplikácie. *AudiometryApp.java* obsahuje vo svojej hlavnej funkcii metódu `launch()`, ktorá volá funkciu `start()` na inicializáciu okna aplikácie a prvej obrazovky.

Trieda *Main.java* existuje na účely kompilácie. Typicky, keď je kód kompilovaný do súboru JAR, je preferované mať triedu *Main.java*, ktorá nededí od žiadnej inej triedy. Preto trieda *Main.java* obsahuje iba hlavnú funkciu, ktorá sa používa na volanie hlavnej funkcie triedy *AudiometryApp.java*.

### 3 Definovanie úrovni hlasitosti

---

Úrovne hlasitosti sa nastavujú v súbore *zvukove\_urovne.config*. Hodnoty úrovni sú uvedené za reťazcom „levels=“ a sú oddelené čiarkou. Tieto hodnoty sú načítané v triede *IndexController.java* a sú premenené na pole hodnôt, ktoré je potom načítané do triedy *Test.java*.



Obr. 3.1: Obsah súboru *zvukove\_urovne.config*.

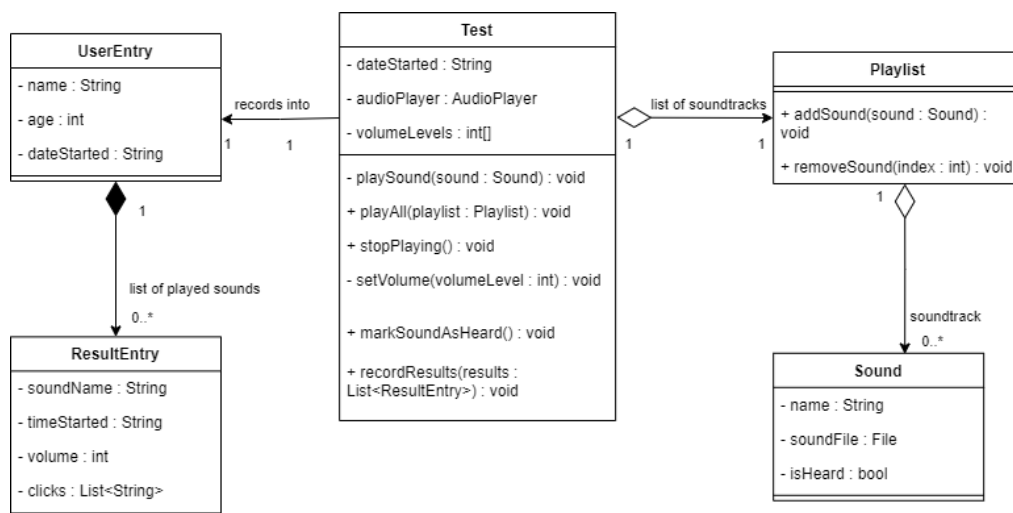


## 4 Testovanie

Táto časť zodpovedá za nasledujúce úlohy v rámci testovania:

- načítavanie zvukových súborov z adresára a ich spravovanie,
- logika testovania - testovacia slučka, vyberanie zvukov na prehrávanie, zmena hlasitosti,
- uchovávanie a zápis výsledkov.

Triedy zahrnuté v tejto časti sú znázornené na konceptuálnom modeli na obrázku 4.1.



Obr. 4.1: UML diagram systému

### 4.1 Zvukové súbory

Zvukové nahrávky použité pri testovaní sa delia do dvoch adresárov *sounds* a *pauses*, ktoré sa nachádzajú v priečinku *src/main/resources*.

Adresár *sounds* obsahuje 9 vybraných zvukov, ktorými sa testuje zvuk používateľa. Tieto nahrávky sú podľa zadávateľa rozdelené do frekvenčných pásiem na nízke, stredné a vysoké.

Adresár *pauses* obsahuje tiché zvukové stopy vo zvukových dĺžkach od 1 po 5 sekúnd, ktoré sú prehrávané ako pauzy medzi zvukmi z adresára *sounds*.

## 4.2 Trieda Sound.java

Trieda *Sound.java* reprezentuje danú zvukovú nahrávku v zdrojovom kóde. Uchováva o nej informácie ako jej názov a cestu k súboru. Tiež obsahuje údaj či na ňu používateľ reagoval.

Keďže pôvodne boli vybrané zvukové nahrávky rozdelené do 3 frekvenčných pásiem, pre triedu *Sound* bol vytvorený enumeračný typ *Frequency.java*, ktorý reprezentuje hodnotu frekvenčného pásma.

## 4.3 Trieda Playlist.java

*Playlist.java* je trieda, ktorá vytvára a spravuje jednotlivé objekty triedy *Sound.java*. Tieto objekty sú rozdelené do dvoch zoznamov *soundtrackList* a *pauseList*, podľa toho či, ide o testovací zvuk alebo tichú nahrávku.

## 4.4 Trieda UserEntry.java

Trieda *UserEntry.java* predstavuje jeden záznam výsledkov testu, ktorý obsahuje informácie o testovanom používateľovi, ako je meno, vek a dátum začiatku testu. Obsahuje zoznam *soundOrder*, ktorý reprezentuje poradie prehrávaných zvukov a tich počas testovania.

## 4.5 Trieda ResultEntry.java

Trieda *ResultEntry.java* reprezentuje jednu položku v zozname *soundOrder* v triede *UserEntry.java*. Jeden objekt *ResultEntry.java* uchováva informácie o nahrávke, ktorá sa v danom čase prehrávala, hlasitosť na ktorej bola prehrávaná, časy reakcií používateľa, a iné.

## 4.6 Trieda Test.java

Trieda *Test.java* je najkomplexnejšia trieda, ktorá prepája všetky ostatné. Zodpovedá za celú testovaciu sekvenciu, implementuje testovaciu logiku a koordinuje objekty.

Test vytvára nový objekt *Playlist.java*, z ktorého vyťahuje zvuky na prehrávanie, a tiež nový objekt *UserEntry.java* na zaznamenávanie výsledkov.

Obsahuje objekt triedy *Clip* z knižnice *javax.sound.sampled* na prehrávanie jednotlivých zvukov.

Trieda *Test.java* obsahuje nasledujúce metódy:

#### **4.6.1 setVolume()**

Metóda, ktorá mení hodnotu hlasitosti zariadenia podľa hodnoty v argumente. Používa na to súbor *nircmd.exe*, ktorý sa nachádza v priečinku *src/main/resources*.

#### **4.6.2 initializeSoundList()**

Táto metóda sa volá na začiatku každého kola testovania, aby sa pripravil nový zoznam zvukov, ktoré sa majú prehrať. Z objektu *Playlist.java* vyberá zvuky a tichá, premieša a usporiada ich v striedavom poradí. Okrem toho zaisťuje, že v zozname budú zahrnuté iba zvuky, ktoré používateľ dovtedy nepočul.

#### **4.6.3 playSound()**

Metóda na prehrávanie jedného zvuku alebo ticha. Volaná ju metódou *playAll()*.

#### **4.6.4 playAll()**

Metóda, ktorá prehráva všetky zvuky a tichá počas testovacích kôl.

#### **4.6.5 markClick()**

Metóda, ktorá označí, že používateľ reagoval na momentálne prehrávaný zvuk.

#### **4.6.6 recordResults()**

Metóda, ktorá po dokončení testu zaznamená všetky údaje z objektu *UserEntry.java* do CSV súboru. Ak tento súbor neexistuje, vytvorí ho.

## 5 Používateľské rozhranie

---

Používateľské rozhranie je vytvorené pomocou knižnice JavaFX. Každá obrazovka je reprezentovaná súborom FXML, ktorý definuje komponenty obrazovky, rozloženie a štýl. Každému súboru FXML je priradený kontrolér, čo je Java trieda, ktorá zabezpečuje funkcionality obrazovky.

Tabuľka 5.1 uvádza zoznam súborov FXML a ich zodpovedajúcich kontrolérov.

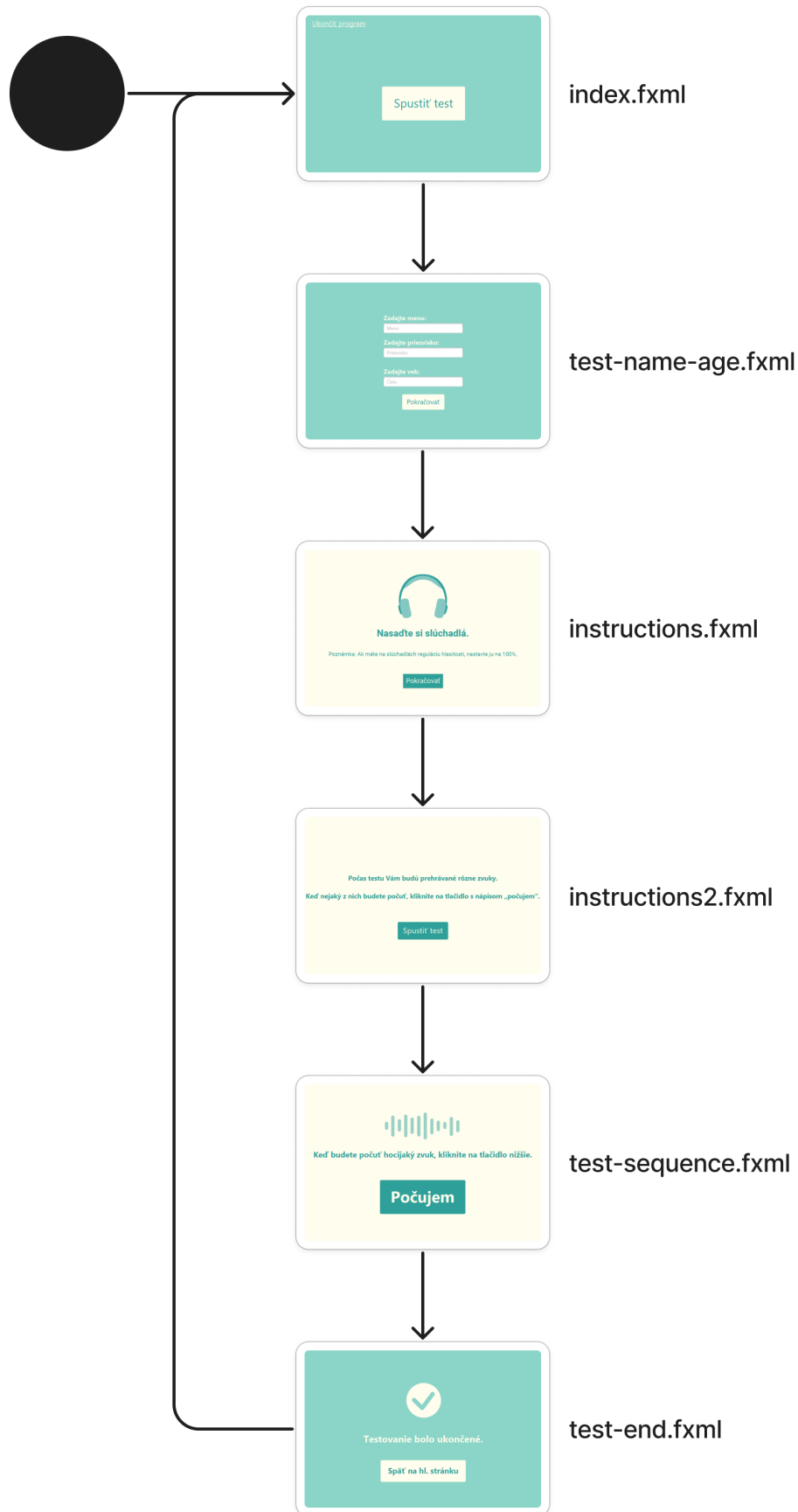
Tabuľka 5.1: Obrazovky rozhrania

FXML súbor	Kontrolér
index.fxml	IndexController.java
test-name-age.fxml	TestNameController.java
instructions.fxml	InstructionsController.java
instructions2.fxml	Instructions2Controller.java
test-sequence.fxml	TestSequenceController.java
test-end.fxml 1.wav	TestEndController.java
alertbox.fxml	AlertBoxController.java

Schéma sekvencie obrazovky je znázornená na obrázku 5.1. Domovská obrazovka *index.fxml* je inicializovaná v triede *AudiometryApp.java* a vytvorí nový objekt triedy *Test.java*. Kontroléry jednotlivých obrazoviek inicializujú nasledujúcu obrazovku v sekvencií a posúvajú si ďalej objekt *Test.java*.

Schéma sekvencie obrazovky neuvádza obrazovku *alertbox.fxml*. Táto obrazovka zobrazuje varovnú správu v prípade, že CSV súbor, do ktorého program zapisuje výsledky, je otvorený inou aplikáciou. Obrazovka *alertbox.fxml* je zobrazená na vyskakovacom okne, ktoré je inicializované triedou *AlertBox.java*.

Sekcia používateľského rozhrania tiež obsahuje sadu ikon umiestnených v priečinku *src/main/resources/icons*, logo aplikácie *appicon.png* a jeden súbor *styles.css*, ktorý bol vytvorený na zmenu štýlu tlačidla v súbore *test-sequence.fxml* podľa aktivity tlačidla.



Obr. 5.1: Diagram postupnosti obrazoviek implementovaného riešenia

## 6 Kompilácia projektu

---

Projekt bol kompilovaný pomocou nástroja Maven. Súbor *pom.xml* obsahuje informácie o projekte a podrobnosti o konfigurácii, ktoré Maven používa na zostavenie projektu do súboru JAR. Aby bola knižnica JavaFX zahrnutá v JAR súbore, do súboru *pom.xml* bol pridaný nasledujúci plugin:

```
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-shade-plugin</artifactId>
<version>2.4.1</version>
<executions>
<execution>
<goals>
<goal>shade</goal>
</goals>
<configuration>
<shadedArtifactAttached>true</shadedArtifactAttached>
<transformers>
<transformer implementation=
"org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">
<mainClass>com.zoramarcinova.audiometry_v3.Main</mainClass>
</transformer>
</transformers>
</configuration>
</execution>
</executions>
</plugin>
```

Pomocou príkazu `mvn install` sa v priečinku `target` vytvorí súbor *Audiometry\_v3-1.0-SNAPSHOT-shaded.jar*.

Aby mal program oprávnenia na zápis do súboru CSV, na spustenie aplikácie bol vytvorený dávkový súbor *Audiometria.bat*. Súbor obsahuje nasledujúce príkazy:

```
@echo off  
java -jar Audiometry_v3-1.0-SNAPSHOT-shaded.jar  
exit
```

Aby program správne fungoval, v jednom priečinku musia existovať tieto tri súbory:

Audiometria.bat,

Audiometry\_v3-1.0-SNAPSHOT-shaded.jar,

zvukove\_urovne.config.