

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Zařízení pro měření reakční doby sportovce

Milan Krejčík

Bakalářská práce

2015

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2013/2014

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Milan Krejčík**  
Osobní číslo: **I11110**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Zařízení pro měření reakční doby sportovce**  
Zadávající katedra: **Katedra informačních technologií**

### **Z á s a d y   p r o   v y p r a c o v á n í :**

Úkolem této práce bude vytvoření softwaru k zařízení pro měření reakční doby sportovce a vyhodnocení výsledků těchto měření.

Zařízení bude tvořeno centrální jednotkou připojenou k PC pomocí USB. K této centrální jednotce jsou napojeny čtyři kontrolní mety, které jsou umístěny okolo sportovce. Na displeji PC nebo laptopu se graficky zobrazuje informace, kterou z těchto met má sportovec dosáhnout. Ten se snaží v co nejkratším čase od zobrazení informace metu doběhnout a nohou sešlápnout.

Výběr mety, ke které bude sportovec mířit je generován náhodně nebo podle nastavené sekvence. Počet pokusů i další parametry by mělo být možno nastavit ve vytvořeném softwaru.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

**HORTON, Ivor. Java 5. Překlad Petr Poledňák. Praha: Neocortex, 2005, 1443 s. ISBN 80-863-3012-5.**

**HEROUT, Pavel. Java: grafické uživatelské prostředí a čeština. 2. vyd. České Budějovice: Kopp, 2007, 316 s. ISBN 978-80-7232-328-9.**

**JARVI, Keane. RXTX. RXTX [online]. 2011 [cit. 2013-11-01]. Dostupné z: <http://rxtx.qbang.org>**

Vedoucí bakalářské práce:

**Ing. Martin Lauterbach**

Katedra elektrotechniky

Datum zadání bakalářské práce:

**20. prosince 2013**

Termín odevzdání bakalářské práce:

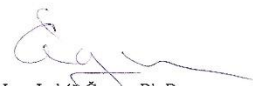
**9. května 2014**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2014

## **Prohlášení autora**

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 2. 9. 2015

Milan Krejčík

## **Poděkování**

Rád bych poděkoval vedoucímu své práce Ing. Martinu Lauterbachovi za cenné rady a velmi vstřícný přístup. Také bych rád poděkoval rodině a svým blízkým přátelům za trpělivost a podporu během mého studia. A především i za finální korekci bakalářské práce, která pro mě byla velice přínosná.

## **Anotace**

Cílem práce je vytvořit software pro měření reakční doby sportovce a vyhodnocení výsledků těchto měření. V teoretické části práce jsou popsány základní pojmy programovacího jazyka Javy, grafické uživatelské prostředí, databáze a sériová komunikace. Praktická část se zabývá návrhem a implementací aplikace pro měření reakční doby sportovce.

## **Klíčová slova**

Java, měření, programování, software, reakční doba, sportovec, meta, SQLite.

## **Title**

The device for measuring the reaction time sportsman.

## **Annotation**

The aim of this thesis is to create software for the measurement of sportsman's reaction time and for evaluation the results of these measurements. Theoretical part describes basic terms of programming language Java, graphical user interface, database and serial communication. Practical part deals with project and implementation of application for the measurement of sportsman's reaction time.

## **Keywords**

Java, measuring, programming, software, reaction time, sportsman, base, SQLite.

## Obsah

<b>Seznam zkratk.....</b>	<b>8</b>
<b>Seznam obrázků.....</b>	<b>9</b>
<b>Seznam zdrojových kódů .....</b>	<b>10</b>
<b>Seznam tabulek.....</b>	<b>11</b>
<b>Úvod .....</b>	<b>12</b>
<b>1 Programovací jazyk Java.....</b>	<b>13</b>
1.1 Prostředí Javy .....	13
1.2 Objekty .....	13
1.3 Třídy .....	14
1.4 Datové složky .....	14
1.5 Metody.....	14
1.6 Zapouzdření .....	14
1.7 Grafické uživatelské rozhraní .....	14
1.7.1 JFrame .....	15
1.7.2 JPanel.....	15
1.7.3 JButton.....	15
1.7.4 JLabel .....	15
1.7.5 JTextField .....	15
1.7.6 JTextArea.....	15
1.7.7 JTable .....	16
1.7.8 Další komponenty z balíčku javax.swing.....	16
1.8 Databáze .....	17
1.8.1 SQLite.....	18
1.8.2 Zprovoznění SQLite .....	18
1.9 Sériová komunikace .....	19
1.10 Java Native Interface.....	20
<b>2 Zařízení.....</b>	<b>21</b>
2.1 Instalace ovladače zařízení .....	21
2.2 Schéma zapojení .....	21
<b>3 Návrh aplikace .....</b>	<b>22</b>
3.1 Popis měření .....	22

3.2	Popis komunikace.....	23
3.3	Návrh GUI.....	23
<b>4</b>	<b>Realizace aplikace.....</b>	<b>24</b>
4.1	Adresářová struktura aplikace .....	25
4.2	Rozvržení aplikace .....	26
4.2.1	Okno připojení zařízení .....	26
4.2.2	Okno měření .....	27
4.2.3	Nabídková lišta .....	28
4.2.4	Nabídková lišta – Nastavení – Sportovci .....	28
4.2.5	Nabídková lišta – Nastavení – Předvolby.....	29
4.2.6	Nabídková lišta – Sekvence.....	31
4.2.7	Nabídková lišta – Výsledky – Vyhodnocení .....	31
4.2.8	Nabídková lišta – Výsledky - Porovnání.....	32
4.2.9	Okno pro sportovce .....	32
4.3	Implementace aplikace .....	34
4.3.1	Balíček sériová komunikace.....	35
4.3.2	Balíček měření.....	35
4.3.3	Balíček sportovci .....	37
4.4	Statistiky.....	40
4.4.1	Platné měření .....	40
4.4.2	Nejdelší reakce .....	40
4.4.3	Nejkratší reakce .....	40
4.4.4	Průměrná reakční doba .....	41
	<b>Závěr.....</b>	<b>42</b>
	<b>Literatura .....</b>	<b>44</b>
	<b>Příloha A – CD s aplikací a zdrojovými kódy.....</b>	<b>45</b>
	<b>Příloha B – ukázka metody getVygenerovanaMeta třídy PomocneMetody .....</b>	<b>46</b>
	<b>Příloha C – ukázka metody rozmerTabulkyPorovnani třídy PomocneMetody.....</b>	<b>48</b>
	<b>Příloha D – ukázka kódu tlačítka pro uložení nastavení ze třídy Predvolby.....</b>	<b>49</b>
	<b>Příloha E – ukázka kódu tlačítka pro uložení sekvence ze třídy Predvolby .....</b>	<b>50</b>
	<b>Příloha F – Příručka uživatele.....</b>	<b>51</b>



## Seznam zkratek

<b>JVM</b>	Java Virtual Machine	Virtuální stroj Javy
<b>GUI</b>	Graphical User Interface	Grafické uživatelské rozhraní
<b>AWT</b>	Abstract Windowing Toolkit	Sada abstraktních nástrojů pro okno
<b>JDBC</b>	Java Database Connectivity	Java připojení k databázi
<b>API</b>	Application Programming Interface	Rozhraní pro programování aplikací
<b>CSV</b>	Comma separated values	Hodnoty oddělené čárkami
<b>URL</b>	Uniform Resource Locator	Jedinečná adresa zdroje
<b>ACID</b>	Atomic, Consistent, Isolated, Durable	Atomičnost, Konzistence, Izolace, Trvanlivost
<b>OS</b>	Operating system	Operační systém
<b>USB</b>	Universal Serial Bus	Univerzální sériová sběrnice
<b>COM</b>	Communication port	Komunikační port
<b>JNI</b>	Java Native Interface	Nativní rozhraní Java

## Seznam obrázků

Obrázek 1 – životní cyklus programu [1] .....	13
Obrázek 2 – architektura JDBC [7] .....	17
Obrázek 3 – schéma zapojení .....	21
Obrázek 4 – zkušební zařízení .....	24
Obrázek 5 – adresářová struktura aplikace .....	25
Obrázek 6 – struktura názvu souboru .....	25
Obrázek 7 – okno připojení zařízení .....	26
Obrázek 8 – okno při neexistujícím sportovci .....	27
Obrázek 9 – okno předvolby .....	30
Obrázek 10 – výběr sekvencí .....	31
Obrázek 11 – vyhodnocení aktuálního měření .....	31
Obrázek 12 – porovnání výsledků .....	32
Obrázek 13 – okno pro sportovce .....	33
Obrázek 14 – struktura balíčků aplikace .....	34

## Seznam zdrojových kódů

Zdrojový kód 1 – třída pro vytvoření nabídkové lišty .....	17
Zdrojový kód 2 – třída pro hledání portu .....	20
Zdrojový kód 3 – vytvoření nativní metody .....	20
Zdrojový kód 4 – výběr správně knihovny dle OS.....	24
Zdrojový kód 5 – ukázka potvrzujícího okna JOptionPane pro smazání sportovce z databáze .....	29
Zdrojový kód 6 – čas na přípravu je náhodně generován z výše zadaného rozmezí.....	29
Zdrojový kód 7 – ukázka nastavení připojení .....	35
Zdrojový kód 8 – ukázka metody pro načtení souboru .....	36
Zdrojový kód 9 – ukázka metody pro načtení sportovce.....	37
Zdrojový kód 10 – ukázka metody pro aktualizaci údajů sportovce.....	38
Zdrojový kód 11 – ukázka metody pro načtení výsledků.....	39
Zdrojový kód 12 – ukázka kódu pro načtení položek do ComboBoxu .....	39
Zdrojový kód 13 – zjištění nejdelší reakce .....	40
Zdrojový kód 14 – zjištění nejkratší reakce.....	40
Zdrojový kód 15 – zjištění průměrné reakční doby .....	41

## **Seznam tabulek**

Tabulka 1 – vyhodnocení přijímaných dat .....	23
---	----

## Úvod

Trénink, zátěžové a výkonnostní testy posouvají sportovce na vyšší dovednostní úroveň. Sportovec potřebuje mít možnost vyhodnotit čas, který věnuje přípravě na sportovní výkony. Analyzovat dlouhé hodiny, dny a roky dřiny. Na základě měření zátěžových, výkonnostních testů může vidět, jestli jeho výsledky se zlepšují a trénink je pro něho přínosný. Aby bylo měření přínosem, musí se provádět ideálně za stejných podmínek. Například běh na stejnou vzdálenost, za stejných podmínek, se stejnou zátěží. Důležitou vlastností je také reakce na určitý povel. Například pro sportovce hrající badminton, je důležité rychle a efektivně reagovat na letící míček. Proto je potřeba, aby sportovec byl neustále v pohybu a včas připraven správně reagovat. Celá problematika se odvíjí od toho, jak rychle je schopný sportovec zareagovat na změnu směru letícího míčku a správně odhadnout jeho dopad.

Cílem této práce je vytvoření aplikace pro měření reakční doby sportovce, která bude využívat zařízení připojené pomocí portu USB. Součástí aplikace bude i modul pro zpracování výsledků měření.

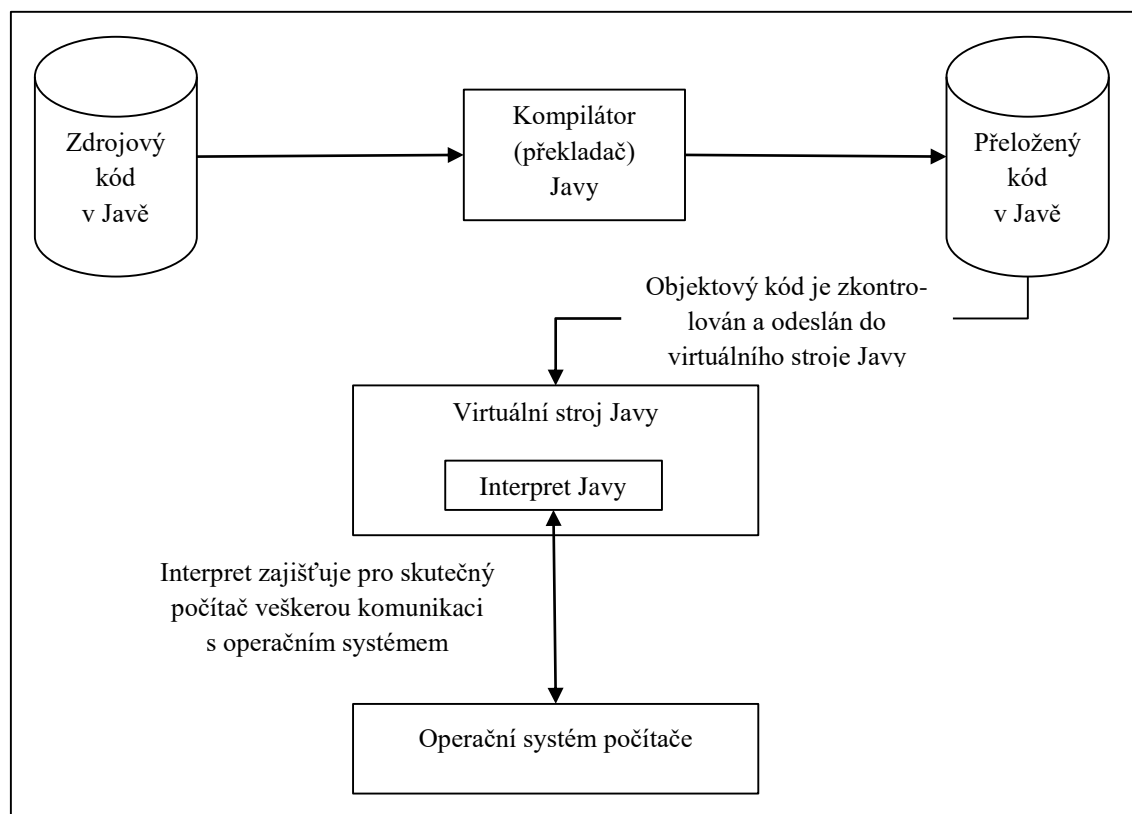
Aplikace by měla být snadno ovladatelná a uživatelsky přívětivá. Pro uživatele intuitivní nastavení parametrů a pro sportovce snadné zobrazení mety, kterou má dosáhnout. Aplikace by mohla informovat uživatele během výkonu sportovce o probíhajícím měření. Výsledkem měření bude vyhodnocení výkonu, které si bude možné prohlédnout v aplikaci, případně ve vytvořeném souboru po skončení měření. Také by aplikace měla umožnit nastavení základních parametrů měření. Základním požadavkem je, aby si uživatel dokázal nadefinovat vlastní sekvenci pohybu sportovce k metám. Ve vlastní sekvenci si určí počet pokusů. U každého pokusu si zvolí metu, která má být sešlápnuta a čas, který bude mít sportovec na sešlápnutí.

# 1 Programovací jazyk Java

Java je moderní programovací jazyk, který byl od počátku vyvíjen jako přenositelný, což znamená, že programy v Javě mohou pracovat velmi podobně bez úpravy zdrojového kódu na jakémkoli operačním systému, který jej podporuje. Program napsaný v Javě vyžaduje jen jedinou sadu zdrojového kódu, na rozdíl od jiných programovacích jazyků. Důležitá vlastnost jazyka je, že je objektově orientován. Objektově orientované programy jsou srozumitelnější a snadnější na údržbu a rozšiřování. [1]

## 1.1 Prostředí Javy

Program v Javě se neprovádí přímo na počítači, ale provádí se na normovaném virtuálním počítači, který se nazývá virtuální stroj Javy neboli JVM a je v počítači emulován programem. Na obrázku č. 1 je znázorněn životní cyklus programu. [1]



Obrázek 1 – životní cyklus programu [1]

## 1.2 Objekty

V Javě je téměř všechno objektem. Jedná se o abstrakci z reality, která obsahuje určitá data. Objekt třídy bude mít danou množinu vlastností společnou a každý objekt bude mít své vlastní hodnoty těchto vlastností. [1]

### 1.3 Třídy

Třída je specifikace, nebo vzor vyjádřený jako část programu, který určuje, čím je vytvořen určitý druh objektu. Třída může mít podtřidu, která zdědí všechny vlastnosti rodičovské třídy, ale také navíc má své specifikace. Instance třídy je technický termín pro existující objekt dané třídy. [1]

### 1.4 Datové složky

Datové složky jsou proměnné pro ukládání datových položek, které zpravidla odlišují jeden objekt třídy od druhého. Dá se na ně odkazovat jako na datové členy třídy. [1]

### 1.5 Metody

Metody definují operace, které můžeme pro třídu provádět – stanovují, co můžeme dělat s objekty dané třídy. Metody zpravidla pracují s datovými složkami. Jedná se o kód programu, který je možné opakovaně volat. Každá metoda má návratový typ. Pokud metoda nevrací žádnou hodnotu, tak se použije návratový typ `void`. K návratu hodnoty z metody se používá příkaz `return`. Dále může mít metoda předávané parametry, které se uvádějí mezi závorkami za jménem metody. [1]

### 1.6 Zapouzdření

Zapouzdření znamená skrytí datových položek a metod uvnitř objektu, pomocí definice třídy jako `private`. Proměnné budou dostupné pouze přes metody definované pro tuto třídu. Zapouzdření je důležité pro bezpečnost a integritu objektů třídy. [1]

### 1.7 Grafické uživatelské rozhraní

Základní prvky, které jsou potřeba k tvorbě GUI, se nacházejí ve dvou balíčcích `java.awt` a `javax.swing`.

- `java.awt`,
  - je hlavním skladem tříd, které se používaly pro tvorbu GUI v Javě od verze 1.1. Komponenty v tomto balíčku jsou do značné míry závislé na nativním kódu, tím pádem jsou omezeny vlastnostmi platformy, na které běží.
- `javax.swing`,
  - tzv. swingové komponenty přidány ve verze 2.0, jedná se o přepracované třídy balíčku `java.awt`, které jsou kompletně napsány v Javě, tím pádem nejsou omezeny vlastnostmi platformy. Komponenty jsou odlišeny začátečním písmenem J.

V této práci jsou častěji použity komponenty z balíčku `javax.swing`. [1] [2]

### 1.7.1 JFrame

Základní třída GUI je `JFrame` pro vytvoření okna. Okno obsahuje rámeček s titulkovou lištou a poskytuje mnoho jiných možností. Aby okno mohlo zobrazit jiné komponenty, musí tato třída dědit ze třídy `Component`, která definuje základní vlastnosti a metody, sdílené všemi komponentami (tuto třídu obsahují všechny swingové komponenty). Další nadtřída je `Container` – definuje komponentu, která může obsahovat jiné komponenty, což je důležitá vlastnost pro `JFrame`. Dále `JFrame` obsahuje nadtřidu `Window` a `Frame`.

### 1.7.2 JPanel

Komponenta `JPanel` představuje něco jako fyzický panel, který se může použít k seskupení více komponent. `JPanel` nám pomáhá lépe rozvrhnout a uspořádat komponenty v našem aplikačním okně.

### 1.7.3 JButton

Třída `JButton` představuje běžné tlačítko. Tlačítko umožňuje provádět předem určenou akci. Po stisknutí se vytvoří `ActionEvent`, což je objekt, který obsahuje informace o události a o jejím zdroji. Aby tlačítko zachytilo například interakci kliknutí myši, musí objekt naslouchat událostem.

### 1.7.4 JLabel

Nejzákladnější textovou komponentou je `JLabel`, která zobrazuje text nebo obrázek. Pokud komponenta neobsahuje žádný text, není nastavené ani pozadí či ikona, tak není na obrazovce vůbec vidět. Komponenta nereaguje na vstupní události a nemůžeme ji editovat, můžeme jí pouze nastavit například akci při kliknutí myši – jedná se tedy o pasivní komponentu. `JLabel` obsahuje mnoho parametrů, které se dají nastavit. Například změna barvy pozadí, velikost písma, jeho zarovnání a jeho barva. Komponenta obsahuje také mnoho metod. Nejpoužívanější jsou metody pro nastavení a získání obsahu.

- Nastavení obsahu – `void setText(String text)`,
- získání obsahu – `String getText()`.

### 1.7.5 JTextField

Komponenta `JTextField` je velmi podobná komponentě `JLabel`. Zobrazuje jeden řádek, ale v jeho případě je editovatelný. Jako u `JLabel` jsou nejpoužívanější metody pro nastavení a získání obsahu. Tyto metody mají stejnou podobu, totéž platí i o jeho parametrech, jenž `JTextField` má nějaké parametry navíc, například parametr, který nám říká, kolik se zobrazí znaků.

### 1.7.6 JTextArea

Do skupiny textových komponent také spadá komponenta `JTextArea`, která je podobná `JTextField`, ale vzhled komponenty je jiný. `JTextArea` umožňuje editovat víceřádkový



text. Text má automatické formátování, pokud je text delší než okno komponenty, tak se automaticky objeví posuvníky, textem lze rolovat i pomocí kurzorových šípek. Komponenta umí zobrazit jak více řádků, tak i více sloupců. Nejuniverzálnější konstruktor je: `JTextArea(String text, int radky, int sloupce)` [2]

### 1.7.7 JTable

Třída `JTable` zobrazuje data ve formě tabulky. Data, která tato komponenta má zobrazovat určuje `TableModel`. Komponenta se především používá pro zobrazení dat z extérní databáze. Položky můžeme vybírat podle řádku, sloupce nebo podle jediné položky. Lze přeskupovat pořadí sloupců, jednoduchým uchopením myši se dá sloupec přetáhnout na novou pozici, čímž dochází k automatické změně pořadí sloupců.

### 1.7.8 Další komponenty z balíčku `javax.swing`

Mezi tlačítkové komponenty můžeme zařadit ještě například `JCheckBox` a `JRadioButton`, které mají parametr `ButtonGroup`, který určuje seskupení těchto objektů. Pokud jsou komponenty seskupeny ve stejné skupině `ButtonGroup`, tak v jednom okamžiku, může být aktivní (stisknutá) jen jedna komponenta. Tyto tlačítka mohou mít textový popis, obrázek nebo obojí.

- `JCheckBox` – zaškrťovací pole,
- `JRadioButton` – přepínač.

Další skupinou jsou kontextové nabídky. Můžeme do svého programu přidat i nabídkovou lištu, která nám pomáhá udělat náš program přehlednější a lépe ovladatelnější.

- `JMenuBar` – nabídkovou lištu prezentuje třída `JMenuBar`, lišta se obvykle nachází v horní části okna. Nelze mít více nabídkových lišt, takzvaně více objektů `JMenuBar`. Pokud se v kódu objeví více objektů `JMenuBar`, tak se použije naposledy přidaný.
- `JMenu` – nabídková položka první úrovně nabídkové lišty. Tato položka se po kliknutí rozvine v seznam nabídkových položek.
- `JMenuItem` – položka v nabídce.

```

public void mojeMenu() {
    JMenuBar nabidkovaLista = new JMenuBar();
    this.setJMenuBar(nabidkovaLista);
    JMenu jMenu1 = new JMenu("Menu1", true);
    nabidkovaLista.add(jMenu1);
    JMenuItem jMenuItem1 = new JMenuItem("Položka1 v Menu1");
    jMenu1.add(jMenuItem1);
    jMenuItem1.addActionListener(this);
    this.setSize(220, 120);
}

```

**Zdrojový kód 1 – třída pro vytvoření nabídkové lišty**

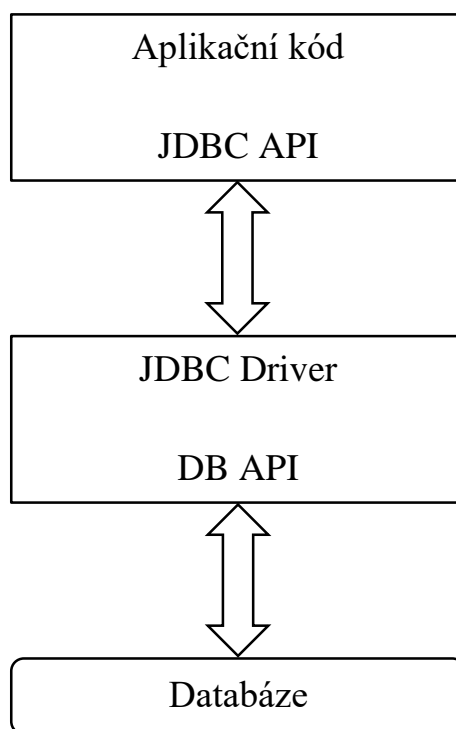
[2]

Nabídková lišta, také může obsahovat `CheckBox` (`JCheckBoxMenuItem`) a `RadioButton` (`JRadioButtonMenuItem`) se stejnými vlastnostmi, viz výše.

## 1.8 Databáze

Java pro práci s databázemi využívá JDBC API. Toto API definuje základní rozhraní nejen pro přístup k relačním databázím, ale třeba i k CSV.

Na obrázku č. 2 je možné vidět základní princip JDBC z pohledu aplikačního programátora.



**Obrázek 2 – architektura JDBC [7]**

K navázání spojení slouží metoda `getConnection()` u třídy `DriverManager` kam se zadá URL. Databázové URL vypadá následovně: „název zdroje“, „login“ a „heslo“. Login

a heslo není vyžadováno, databáze může být bez přihlašovacích údajů. Např. pro SQLite bez přihlašovacích údajů vypadá metoda takto: `getConnection("jdbc:sqlite:DB.db")`.

Po spojení s databází, můžeme začít vykonávat operace, které popisuje třída `Statement`. Instance této třídy posílá SQL příkazy databázi, kde jsou zpracovány a vrací odpovídající hodnotu. Nejzákladnější metody třídy `Statement` jsou:

- `executeQuery()` – slouží k posílání dotazů `SELECT` do databáze, navrácí objekt třídy `ResultSet`, který již obsahuje skutečná data a s nimiž můžeme dále pracovat;
- `executeUpdate()` – slouží k aktualizaci databáze (`UPDATE`, `INSERT`, `DELETE`).

[6] [7]

### 1.8.1 SQLite

Jedná se o malou knihovnu, napsanou v programovacím jazyce C. Knihovna je dostupná pro operační systémy:

- Windows,
- Linux,
- Mac OS X.

Výhodou je, že tato databáze není založená na principu server-klient, tudíž nemusí být spuštěn samostatný proces databázového serveru. Kód SQLite je v „public domain“, což znamená, že je zdarma šířitelná, jak pro komerční nebo soukromé účely. Stačí, když se knihovna přilinkuje k aplikaci. SQLite si vystačí s jednoduchým rozhraním. Samotná databáze je uložena v souboru s příponou `db`. Formát souboru je multiplatformní a nezáleží na tom, zda byl vytvořen v 32-bitovém nebo 64-bitovém systému. SQLite je například vestavěná v mobilním operačním systému Android. Samozřejmostí je podpora ACID:

- atomičnost – dotaz se buď provede celý nebo se neprovede vůbec,
- konzistence – při provádění dotazu není porušeno žádné integritní omezení,
- izolace – operace uvnitř transakce jsou skryty před vnějšími operacemi,
- trvanlivost – úspěšně provedená změna se uloží v databázi a již se nedá ztratit.

SQLite obsahuje téměř celý standard SQL-92 a některé nadstandardní prvky, ale neobsahuje například:

- spojování výsledků dotazu `SELECT` – `OUTER RIGHT JOIN`,
- odebrání sloupce v tabulce – `ALTER TABLE DROP COLUMN`,
- nastavení/odebrání přístupových práv – `GRANT/REVOKE`.

[8]

### 1.8.2 Zprovoznění SQLite

Před používáním SQLite v Java aplikaci je nutné připojit k aplikaci knihovnu SQLite JDBC Driver. Připojení k databázi realizujeme výše popsáním postupem. Pokud soubor `.db`

uvedený ve zdroji neexistuje, tak se automaticky vytvoří. Po úspěšném vytvoření databáze a následném připojení se mohou začít realizovat SQL dotazy. Základní kostra databáze se může vytvořit dotazy předané v metodě `executeUpdate()` ve třídě `Statement`. Je nutné upozornit na skutečnost, že SQLite má automaticky nastavený auto commit (automatické uložení výsledku dotazu). Někdy se tato vlastnost může hodit, někdy se naopak hodit nemusí, proto je často využívaná metoda `setAutoCommit()` s atributem `false` a samotný commit se provede metodou `commit()` ve třídě `Connection`. Nemělo by se ani zapomínat na uvolňování prostředků metodou `close()` ve třídě `Statement` a taktéž ve třídě `Connection`. Pro výpis záznamů v tabulce můžeme zvolit výpis do konzole nebo swingovou komponentu `JTable` popsanou v kapitole 1.7.7. V obou případech se pro výpis použije `while` cyklus za použití iterátoru ve třídě `ResultSet`. Mohou se provádět operace:

- `Insert,`
- `Select,`
- `Update,`
- `Delete.`

A samozřejmě další operace popsané ve standardu SQL-92. [9]

## 1.9 Sériová komunikace

Standardní knihovnou pro sériovou komunikaci je knihovna Sun JavaComm, která je ale k dispozici jen pro operační systémy Solaris/SPARC, Solaris/x86 a Linux x86. Pro Windows oficiálně není dostupná. [4] Pro Windows existuje alternativní knihovna RXTX. Knihovna RXTX je vydána ve dvou verzích, pro 32-bitový a 64-bitový operační systém. Proto je nutné zvolit správnou verzi knihovny. Ke knihovně se váže ještě jedna knihovna – `rxtxSerial.dll`, která je taktéž vydána ve verzi pro 32-bitovou nebo 64-bitovou verzi operačního systému. Ta musí být v adresáři současně s aplikací. [5] Jak knihovna JavaComm, tak RXTX poskytují podobné možnosti pro práci se sériovým portem:

- umožňuje získání seznamu dostupných portů,
- nastavení parametrů připojení,
- komunikace – přenos dat ze zařízení a do zařízení.

RXTX je Javovská knihovna, používající nativní implementaci (JNI), poskytující sériovou a paralelní komunikaci pro Javu. Knihovna je založena na specifikaci od Sun JavaComm.

```

public void hledaniPortu() {

    guiSerialPort.jComboBoxCom.removeAllItems();

    porty = CommPortIdentifier.getPortIdentifiers();
    while (porty.hasMoreElements()) {
        CommPortIdentifier aktualniPort =
            CommPortIdentifier(porty.nextElement());

        if (aktualniPort.getPortType() ==
            CommPortIdentifier.PORT_SERIAL) {
            guiSerialPort.jComboBoxCom.addItem(aktualniPort
                .getName());
            portMap.put(aktualniPort.getName(),
                aktualniPort);
        }
    }
}

```

**Zdrojový kód 2 – třída pro hledání portu**

[3]

## 1.10 Java Native Interface

JNI je rozhraní, které umožňuje propojit kód běžící na virtuálním stroji Javy s nativními programy a knihovnami napsanými v jiných jazycích. JNI se využívá především pro komunikaci s hardwarovými zařízeními, ale také i když je potřeba využít starší nativní kód. Volání v aplikaci se realizuje při vytvoření nativní metody, která se poté volá stejným způsobem jako jakákoli jiná metoda napsaná v Javě. Na pozadí dochází prostřednictvím JNI ke spuštění metody, jejíž implementace je napsána v jiném jazyce a je uložena v nativní knihovně. Hlavní nevýhodou JNI je, že se při použití JNI ztrácí podpora nezávislosti na platformě. Kód v Javě bude stále přenositelný, ale nativní knihovny musejí být znovu přeloženy pro každý operační systém zvlášť.

```

private native void nativniMetoda();

```

**Zdrojový kód 3 – vytvoření nativní metody**

[10]

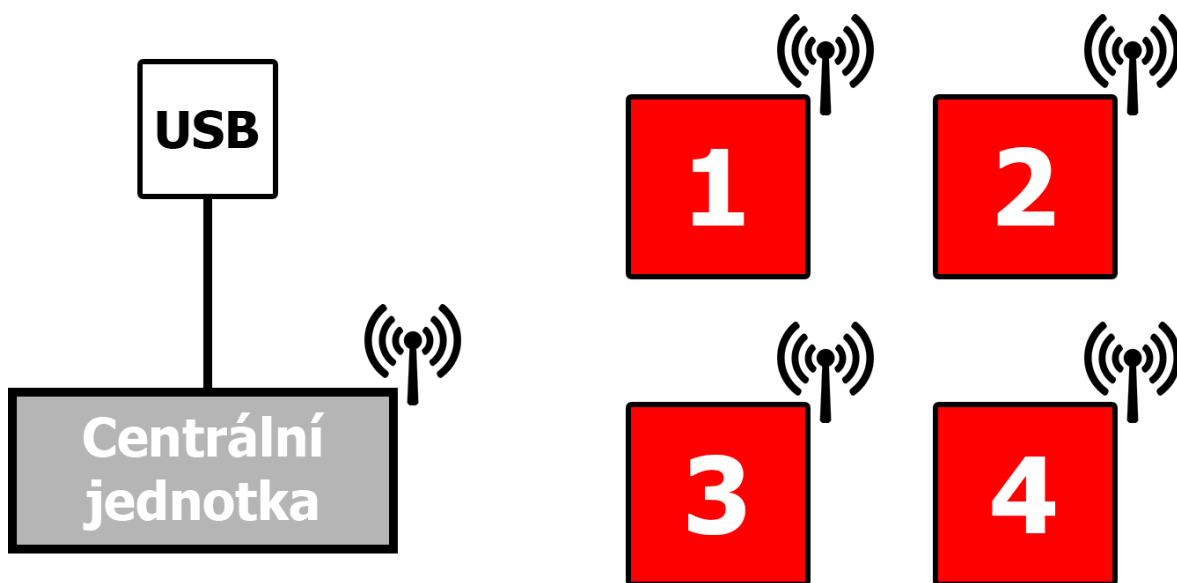
## 2 Zařízení

Zařízení je tvořeno centrální jednotkou připojenou k PC pomocí USB za využití Virtual Com Portu (OS rozpozná zařízení, jako kdyby bylo připojené ke COM portu). Centrální jednotka komunikuje bezdrátově v pásmu 2.4 GHz se čtyřmi bezdrátovými kontrolními mety, které jsou napájeny baterií. Mety nepotřebují externí zdroj, tím je zajištěna bezpečnost pro sportce při vykonávání měření, kde by hrozilo zakopnutí o kabeláž. Součástí firmwaru mety je její jednoznačné označení, toto označení je vytištěné ze spodní strany. Při instalaci met je nutné dodržet správné rozmístění. Rozmístění je znázorněno ve schématu zapojení (Obrázek 3 – schéma zapojení). Mety mohou být vzdáleny od sportovce maximálně 10 metrů, z důvodu vlastností použitého vysílače.

### 2.1 Instalace ovladače zařízení

Jako každé USB zařízení potřebuje zařízení pro měření reakční doby sportovce nainstalovat ovladač do PC. Aktuální ovladač dodává výrobce zařízení a je možné ovladač stáhnout a nainstalovat z jejich webových stránek. U některých ovladačů je také potřeba mít povolené v operačním systému instalovat nepodepsané ovladače.

### 2.2 Schéma zapojení



Obrázek 3 – schéma zapojení

### 3 Návrh aplikace

Aplikace bude vyvíjena s ohledem na snadné použití s jednoduchým uživatelským rozhraním. K naprogramování aplikace byl zvolen objektově orientovaný programovací jazyk Java vyvinut firmou Sun Microsystems. Java je nejpoužívanější programovací jazyk na světě. Je oblíben především svojí jednoduchostí a snadnou přenositelností mezi platformami.

Návrh aplikace bude postupovat systematicky od připojení zařízení k PC, následně dojde k navázání spojení se zařízením s aplikací, kde bude docházet k zachytávání dat. V hlavním okně aplikace se budou zobrazovat důležité informace o průběhu měření, jako je celkový čas měření, počet pokusů, kolik pokusů sportovci zbývá a o jakého sportovce se jedná. Dále v tomto okně bude možné sledovat aktuální vyhodnocení pokusů, jaká byla sešlápnuta meta, neboli zda byl daný pokus platný. V tomto okně se bude nacházet nabídková lišta, která bude rozcestníkem pro další volby. Z tohoto menu bude možné otevřít okno s nastavením pro měření a také třeba zobrazit databázi sportovců. Dále půjde vybrat požadovanou sekvenci generování met, která bude určovat průběh měření sportovce. Z nabídky půjde ještě vyvolat okna pro vyhodnocení výsledků sportovce.

Okno sportovců bude zobrazovat základní identifikační údaje sportovce a bude možné zobrazit veškeré výsledky daného sportovce. Zde nebude chybět funkcionalita pro přidání, úpravu a odebrání sportovce z databáze.

V okně předvoleb bude umožněno uživateli nastavit vlastní sekvenci či přenastavit výchozí parametry měření.

V oknech vyhodnocení výsledků bude umožněno zobrazit aktuálně ukončené měření, kde se budou nacházet informace o výsledku měření, zda bylo v pořádku či nikoli, což se bude vyhodnocovat na základě toho, zda sportovec sešlápl všechny požadované mety v požadovaném čase. A základní statistické vyhodnocení nejdelší, nejkratší a průměrná reakční doba. Poté půjde tyto výsledky porovnávat se svými výsledky či s výsledky jiného sportovce.

Samotné měření bude prováděno v okně, které bude maximalizované na obrazovku PC. V tomto okně budou pouze zobrazené mety a na jednotlivé metě bude graficky znázorněné, která má být právě sešlápnuta.

#### 3.1 Popis měření

Náhodně se vygeneruje přípravný čas, od zobrazení mety začne běžet odpočet (timeout), po sešlápnutí mety se zaznamená reakční čas (pouze v případě, pokud byla sešlápnuta správná meta před vypršením timeoutu).

Výsledky měření je ukládáno do csv souboru a také do databáze.

### 3.2 Popis komunikace

Zařízení neustále posílá informace o svém stavu. Ve výchozím stavu – nesešlápnutá žádná meta, posílá hodnotu 0x00. V případě sešlápnutí více met naráz zařízení pošle odpovídající data, ale aplikace tyto data vyhodnotí jako chybné. Po sešlápnutí mety dojde k příjmu dat a následnému vyhodnocení. V případě zaslání jiných dat ze zařízení budou data vyhodnocena jako chybná.

Přijímaná data	Význam
0x00	Nesešlápnuta žádná meta
0x01	Sešlápnuta meta 1
0x02	Sešlápnuta meta 2
0x04	Sešlápnuta meta 3
0x08	Sešlápnuta meta 4

Tabulka 1 – vyhodnocení přijímaných dat

### 3.3 Návrh GUI

Uživatel bude schopný spustit a vyzkoušet aplikaci i bez připojeného zařízení pro měření reakční doby sportovce. Po spuštění aplikace tedy bude vyzván k připojení zařízení a až poté se zobrazí hlavní okno aplikace. V hlavním okně aplikace bude možné sledovat průběh měření a také z nabídkové lišty zobrazit nastavení měření. Měření bude prováděno v samostatném okně, které bude maximalizované na obrazovce PC.



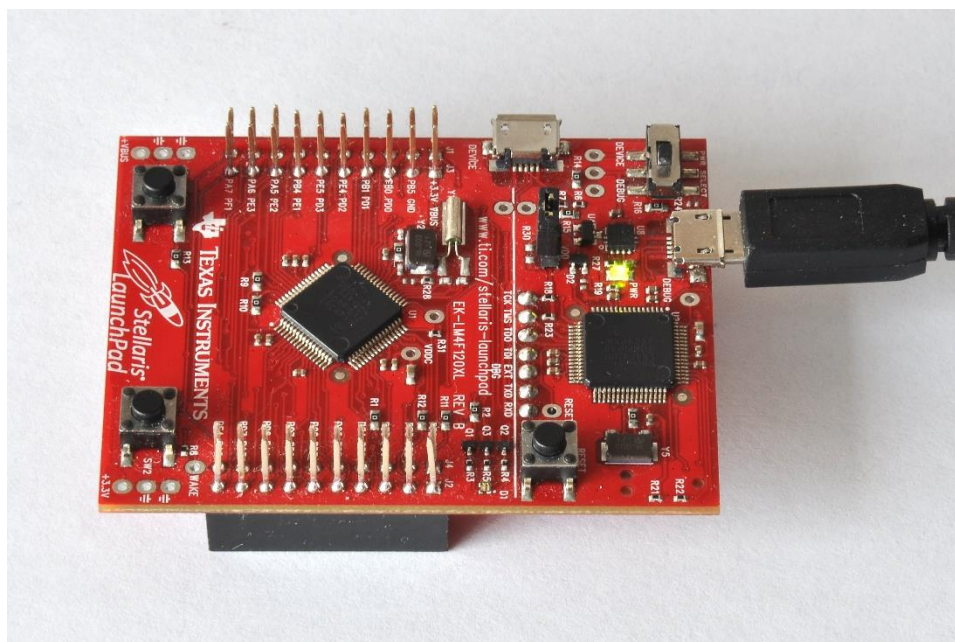
## 4 Realizace aplikace

Aplikace je naprogramována v programovacím jazyce Java. Aplikace umí zjistit verzi operačního systému Windows a podle toho zvolit správnou knihovnu pro sériovou komunikaci, která je vydána ve verzi pro 32-bit nebo pro 64-bit operační systém. Nebylo tedy nutné aplikaci vydávat ve dvou verzích. Pro správné fungování aplikace je nutné mít nainstalovanou aktuální verzi Javy.

```
if (System.getProperty("sun.arch.data.model").equals("32")) {  
    System.setProperty("java.library.path", "data/x32");  
} else {  
    System.setProperty("java.library.path", "data/x64");  
}  
Field fieldSysPath =  
    ClassLoader.class.getDeclaredField("sys_paths");  
fieldSysPath.setAccessible(true);  
fieldSysPath.set(null, null);
```

**Zdrojový kód 4 – výběr správné knihovny dle OS**

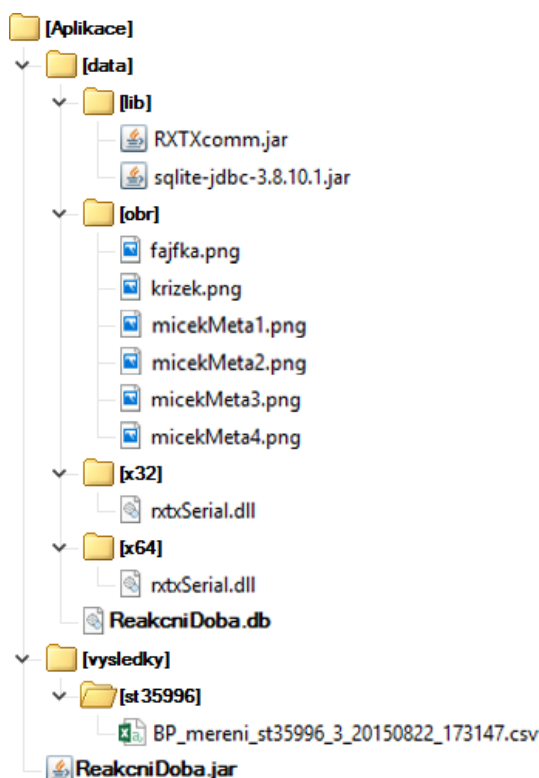
Pro testování aplikace byla použita náhrada v podobě kitu a přenos dat byl ekvivalentní se skutečným zařízením. Kvůli omezení zařízení nebylo především možné provést ostrý test se sportovcem.



**Obrázek 4 – zkušební zařízení**

## 4.1 Adresářová struktura aplikace

Aplikace má pevně danou adresářovou strukturu. Ve složce data se nacházejí důležité soubory pro správný chod aplikace a nesmí zde být nic upravováno. Složka lib obsahuje knihovnu pro sériovou komunikaci RXTX a pro databázi SQLite. V adresáři obr jsou příslušné obrázky, které se vyskytují v aplikaci. Pro správný chod sériové komunikace jsou zde složky „x32“ a „x64“ ve kterých je uložena knihovna „rxtxSerial.dll“ (rozdílná podle OS). Aplikace sama pozná, o jaký systém se jedná a podle toho si pro knihovnu sáhne do správného adresáře. Dále je zde databázový soubor „ReakcniDoba.db“, který je umístěn v kořenu adresáře data. Databázový soubor nemusí být součástí aplikace a databáze se po spuštění aplikace případně vytvoří. To samé platí i pro složku výsledky, která se automaticky vytvoří při uložení výsledku měření v kořenovém adresáři aplikace, v případě, že zde složka nebude existovat. V kořenovém adresáři se ještě nachází samotný spouštěcí soubor „ReakcniDoba.jar“.



Obrázek 5 – adresářová struktura aplikace

Soubor s uloženými výsledky měření má podobu:

**BP\_mereni\_st35996\_3\_20150830\_110118.csv**

Bakalářská práce\_měření\_Net ID\_počet pokusů\_rrrrmmdd\_hhmmss.csv

Obrázek 6 – struktura názvu souboru

## 4.2 Rozvržení aplikace

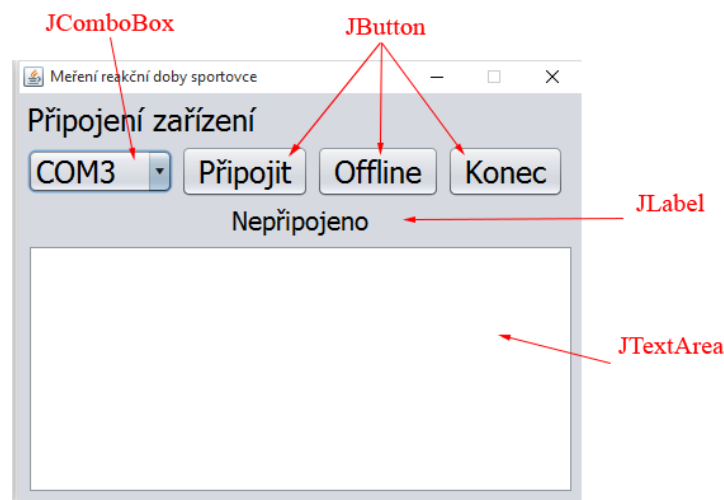
Aplikace je rozvržena do více oken, které na sebe logicky navazují nebo se vyvolají příslušnou akcí. Aplikace disponuje čtyřmi důležitými parametry. První parametr je **timeout**, jedná se o čas, který bude mít sportovec na sešlápnutí mety. Další časové parametry jsou **dolní a horní mez přípravy**. Z tohoto intervalu je náhodně generovaný čas pro přípravu sportovce před zobrazení další mety. Časy timeout, dolní mez přípravy a horní mez přípravy jsou zadávány v milisekundách. Poslední parametr je **počet pokusů**, tzv. kolik má sportovec sešlápnout met v měření.

Parametry jsou nastaveny ve výchozím nastavení na hodnoty:

- timeout – 5 000 ms,
- počet pokusů – 5,
- dolní mez přípravy – 2 000 ms,
- horní mez přípravy – 3 000 ms.

### 4.2.1 Okno připojení zařízení

Po spuštění aplikace se zobrazí okno pro připojení zařízení. Okno má pevně nastavenou velikost a nedá se měnit. Další vlastnost, které toto okno má nastavené je, že bude vždy navrchu. Jak bylo zmíněno, aplikace se dá vyzkoušet i bez připojení zařízení. V rozbalovacím menu (JComboBox) se vybere port, na kterém je zařízení připojené. Jestliže není připojené žádné zařízení, které využívá COM port, tak je tento seznam prázdný. Pokud dojde k připojení zařízení až po spuštění aplikace, tak po kliknutí na rozbalovací menu dojde k aktualizaci COM portů. Okno dále obsahuje tři tlačítka (JButton), nálepku (JLabel) a textové pole (JTextArea). Po stisknutí tlačítka „Připojit“ dojde k připojení zařízení. Tlačítko „Offline“ slouží pro vyzkoušení aplikace bez připojeného zařízení. Tlačítko „Konec“ ukončí aplikaci. V JLabelu se zobrazuje aktuální informace zařízení v jakém stavu se nachází. Do textového pole se po připojení zařízení vypisuje aktuální informace, které neustále zařízení posílá.



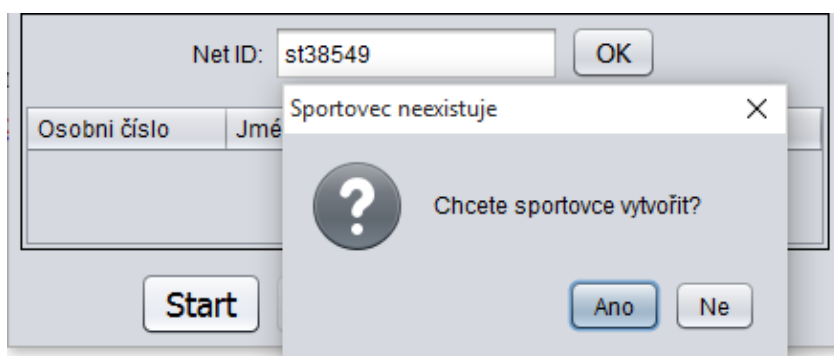
Obrázek 7 – okno připojení zařízení

## 4.2.2 Okno měření

V tomto okně se zobrazují veškeré informace potřebné k měření. Nachází se zde nabídková lišta (JMenuBar). Lišta obsahuje čtyři nabídky (JMenu).

- Nastavení – nabídka obsahující položku Sportovci a Předvolby.
  - Sportovci – databáze sportovců – viz kapitola 4.2.4.
  - Předvolby – nastavení měřících parametrů – viz kapitola 4.2.5.
- Sekvence – na výběr je pár přednastavených sekvencí, sekvence jsou popsány v kapitole 4.2.6.
- Výsledky – po skončení měření se dá nejenom vyhodnotit aktuálně skončené měření, ale také jej porovnat s předchozími výsledky. Viz kapitola 4.2.7 a 4.2.8.
- Nápověda – obsahuje položku „o aplikaci“.

V tomto okně se dále zobrazuje aktuální datum a čas. Dalším časovým údajem je celkový čas měření. S tímto časem souvisí čas nazvaný Timeout (popsaný v kapitole 4.2.5). Dále se zobrazuje údaj o počtu provedených pokusů a počet zbývajících pokusů. Důležitou sekci je zadání Net ID Sportovce pro přiřazení výsledků ke správnému sportovci. Před zahájením měření je nutné zadání Net ID potvrdit tlačítkem OK. Jestliže bude sportovec existovat, vypíše se jeho osobní číslo, jméno, příjmení a rok narození. Pokud neexistuje, pak se tyto informace nevypíší a při stisknutí tlačítka Start se aplikace dotáže, jestli se má sportovce vytvořit.



Obrázek 8 – okno při neexistujícím sportovci

Bez vytvoření sportovce nelze měření zahájit. Poslední sekci v okně je sekce tlačítek (JButton).

- Start – dojde k otevření okna pro sportovce na celou obrazovku, aby sportovce nic nerušilo a mohl se plně soustředit na svůj výkon. Pokud není zadáno Net ID, aplikace na to upozorní a okno se neotevře. Tlačítko se zpřístupní až po zadání sportovce.
- Reset – po stisku se vynulují všechny hodnoty do výchozího (nastaveného) nastavení. Tlačítko se zpřístupní až po absolvování měření.
- Uložit – slouží k uložení výsledků do souboru ve formátu csv. Tlačítko se zpřístupní až po absolvování měření.
- Konec – ukončí celou aplikaci se všemi otevřenými okny.

Pokud se používá více jak jeden monitor a uživatel chce při měření kontrolovat stav v tomto okně, je nutné toto okno přetáhnout na druhý monitor před stisknutím tlačítka Start. Okno pro sportovce se automaticky otevře na celou obrazovku monitoru, který je nastaven jako hlavní monitor. Tlačítka Start, Reset, Uložit a Konec jsou opatřeny potvrzujícím dialogem.

### 4.2.3 Nabídková lišta

Nabídková lišta obsahuje nabídky (JMenu) s položkami (JMenuItem).

- Nastavení – nabídka obsahující položku Sportovci a Předvolby.
  - Sportovci – databáze sportovců – viz kapitola 4.2.4.
  - Předvolby – nastavení měřících parametrů – viz kapitola 4.2.5.
- Sekvence – na výběr je pár přednastavených sekvencí, sekvence jsou popsány v kapitole 4.2.6.
- Výsledky – po skončení měření se dá vyhodnotit aktuálně skončené měření, a také jej porovnat s předchozími výsledky. Viz kapitola 4.2.7 a 4.2.8.
- Nápověda – obsahuje položku „o aplikaci“.

### 4.2.4 Nabídková lišta – Nastavení – Sportovci

První položka v nabídce nastavení je položka Sportovci.

Při otevření okna se sportovci dojde k dotazu `SELECT * FROM Sportovci` z databáze a výsledek se vypíše na JTable, který je rozdělen na příslušné sloupce a řádky, do kterých se načtou příslušné hodnoty. Načítají se data ze sloupců: ID, NetID, Osobní číslo, Jméno, Příjmení, Rok narození, Pohlaví, Typ studia, Ročník, Fakulta, Výška, Váha, Sport a nejlepší dosažený výsledek (celkový čas po dobu měření).

- ID – automatické nastavení ID sportovce podle přidání do databáze,
- Net ID – jednoznačná identifikace sportovce,
- Sport – zda sportovec vykonává aktivně nějaký sport, jestliže ano – zadat jaký sport,
- Výsledek – dosažený nejlepší celkový výsledek měření.

Pro rychlejší nalezení sportovce je zde možnost vyhledání podle Net ID. Text „Sportovec nevybrán“ upozorňuje, že nejdou dělat operace nad sportovcem. Tím je myšleno – upravit, smazat a zobrazit výsledky sportovce. Po kliknutí na tato tlačítka dojde k upozornění vyskakovacím oknem, že se nejdříve musí vybrat sportovec. Je tedy nutné nejdříve kliknout na řádek sportovce a tím dojde k jeho vybrání, text „sportovec nevybrán“ se změní na již vybraného sportovce. Lišta ovládání obsahuje tato tlačítka:

- Přidat sportovce – zobrazí se okno pro přidání nového sportovce,
- Upravit sportovce – slouží k editaci sportovce,
- Smazat sportovce – vyskočí potvrzující dialog o smazání, po potvrzení dojde ke smazání,
- Výsledky sportovce – prohlížení výsledků.

```

String[] nastaveni = {"Ano", "Ne"};
int volba = JOptionPane.showOptionDialog(null,
    "Opravdu smazat sportovce?", "Potvrďte",
    JOptionPane.YES_NO_OPTION,
    JOptionPane.QUESTION_MESSAGE, null, nastaveni,
    nastaveni[0]);
if (volba == 0) {
    Statement stmt = null;
    try {
        stmt = conn.createStatement();
        String sql = "DELETE FROM Sportovec WHERE id
            LIKE " + id;
        System.out.println(sql);

        stmt.executeUpdate(sql);
        nactiSportovce();
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this,
            "Sportovec nebyl smazán.");
    }
}
}

```

**Zdrojový kód 5 – ukázka potvrzujícího okna JOptionPane pro smazání sportovce z databáze**

Při smazání sportovce nedochází k smazání zaznamenaných měření, z důvodu nechtěného smazání. Výsledky je poté nutné z počítače smazat ručně. O evidenci se stará databázový soubor ReakcniDoba.db – nedoporučuje se tento soubor upravovat jiným způsobem než přímo v aplikaci.

#### 4.2.5 Nabídková lišta – Nastavení – Předvolby

Druhou položkou v nabídce Nastavení je položka Předvolby, jedná se o okno pro nastavení parametrů měření. Po otevření okna je vidět jen prázdná tabulka (JTable) a přepínač (JRadioButton) nastavení parametrů. Přepínač je v poloze „Výchozí“, to znamená, že parametry jsou převzaty z výchozího nastavení – viz kapitola 3. Parametry k nastavení se zobrazí, až když se klikne na „nastavit“:

- timeout – čas na sešlápnutí mety (zadáváno v milisekundách),
- počet pokusů – kolik pokusů bude mít sportovec na sešlápnutí,
- dolní mez přípravy – určuje první hodnotu z času na přípravu,
- horní mez přípravy – určuje druhou hodnotu z času na přípravu.

```

long pripravnyCas = rand.nextInt(horniMezPripravy -
    dolniMezPripravy) + dolniMezPripravy;

```

**Zdrojový kód 6 – čas na přípravu je náhodně generován z výše zadaného rozmezí**

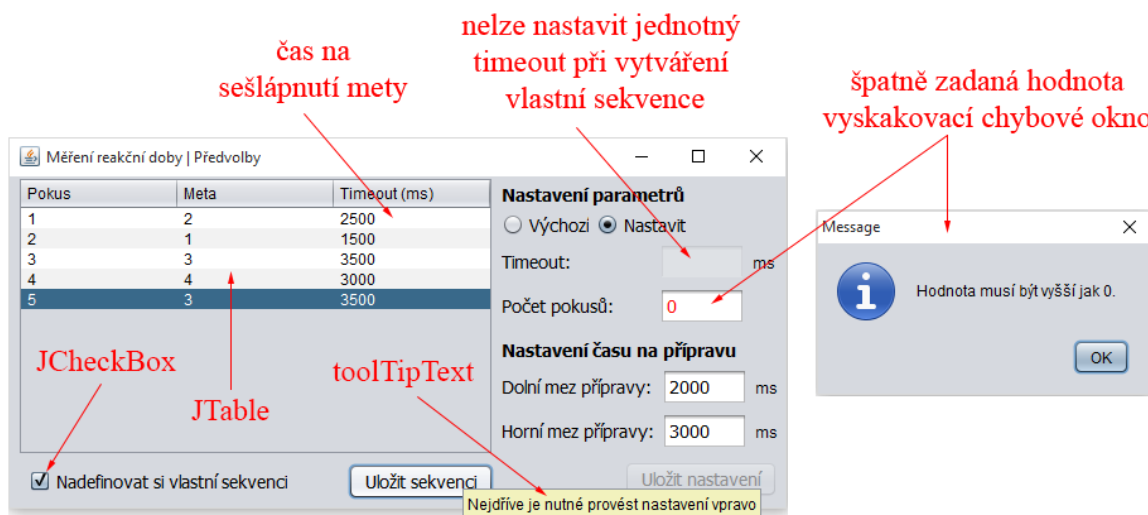
Nastavení je nutné poté uložit stisknutím tlačítka „Uložit nastavení“. Vrátit se k výchozím hodnotám je jednoduché – stačí kliknout zpět na „Výchozí“ a uložit nastavení.

Aplikace upozorní na chybné zadání. První upozornění nastane, když se do pole zadá špatná hodnota – špatná hodnota zčervená. Následuje druhé upozornění – zobrazení vyskakovacího okna s chybovou hláškou. Ke třetímu upozornění dojde, pokud se bude ignorovat červeně zvýrazněné písmo s chybovou hláškou, pokud se stiskne tlačítko uložení. Vyskakovací okno se nezobrazí v případě, že budou špatně nastaveny meze času k přípravě – text pouze zčervená, z důvodu přívětivějšího ovládání.

Vlastní sekvence se nadefinuje po kliknutí na „Nadefinovat si vlastní sekvenci“. Načte se tabulka s počtem pokusů, který je zadán (výchozí hodnota pro pokusy je 5). Tím dojde ke zneaktivnění tlačítka „Uložit nastavení“. Takže pro změnu počtu pokusů je nejdříve nutné odškrtnout „Nadefinovat si vlastní sekvenci“. Tabulka se nastavuje následovně:

- sloupec „Meta“ – číslo mety, která bude zobrazena pro sešlápnutí,
- sloupec „Timeout“ – čas v milisekundách, který udává, kolik času bude mít sportovec na sešlápnutí mety.

Pro uložení sekvence musí být vyplněny všechny hodnoty a poslední hodnota potvrzena klávesou Enter. Po uložení sekvence, dojde k výběru sekvence v nabídkové liště v hlavním okně – viz kapitola 4.2.6. Když bude odškrtnuté, že se nechce nastavovat sekvence, nastaví se v nabídce Sekvence na sekvenci náhodnou.

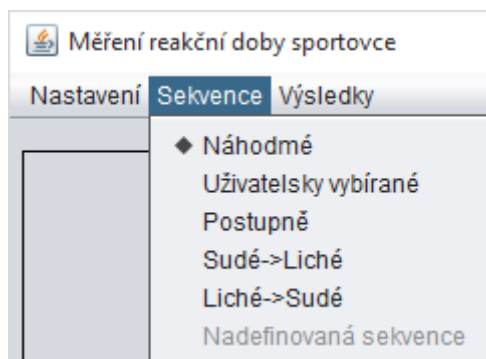


Obrázek 9 – okno předvolby

Je třeba si pořádně promyslet sekvenci. Do již vytvořené sekvence nejdou vkládat a odebírat pokusy, lze jen editovat již předem daný počet pokusů. Nastavovaný čas je v milisekundách.

#### 4.2.6 Nabídková lišta – Sekvence

Nabídka (JMenu) Sekvence nabízí na výběr nadefinované sekvence.



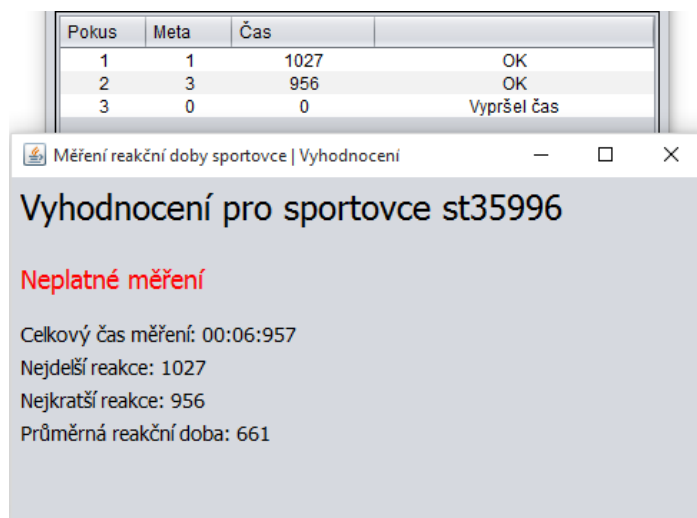
Obrázek 10 – výběr sekvencí

- Náhodné – mety se budou zobrazovat náhodně,
- Uživatelsky vybírané – před každým zobrazením mety se aplikace dotáže jaká meta má být zobrazena, metu si zvolíte v zobrazeném vstupním dialogu,
- Postupně – mety se budou zobrazovat postupně v pořadí: 1,2,3,4,1 atd.,
- Sudé->Liché – mety se budou zobrazovat v pořadí: 2,4,1,3,2 atd.,
- Liché->Sudé – mety se budou zobrazovat v pořadí: 1,3,2,4,1 atd.,
- Nadefinovaná sekvence – tato volba je z této nabídky neaktivní, lze aktivovat jen nastavením vlastní sekvence v předvolbách, viz kapitola 4.2.5.

#### 4.2.7 Nabídková lišta – Výsledky – Vyhodnocení

Po skončení měření je důležité vyhodnocení. Jedná se o okno, v kterém je zobrazeno vyhodnocení aktuálního skončeného měření. Položka ve výchozím nastavení je neaktivní, po skončení měření se zaktivní a lze si prohlédnout vyhodnocení.

Zobrazují se zde informace, zda proběhlo měření v pořádku, celkový čas měření, nejdelší reakci, nejkratší reakci a průměrnou reakční dobu.



Obrázek 11 – vyhodnocení aktuálního měření



#### 4.2.8 Nabídková lišta – Výsledky - Porovnání

Porovnání výsledků je důležitá vlastnost. Aby měření mělo nějaký význam, musí se porovnávat výsledky a zjišťovat zda dochází ke zlepšení. To vše je k zobrazení v tomto okně. Na výběr jsou sportovci, kteří jsou zaneseni v databázi. Takže na výběr bude i sportovec, který ještě neabsolvoval žádné měření. Sportovec, který již měření absolvoval, tak po jeho výběru je umožněno vybrat konkrétní měření. Poté se výběr potvrdí tlačítkem „Načíst výsledky“, nebo stiskem mezerníku. V tabulce se zobrazí výsledky a pod ní souhrnné informace. Lze porovnávat své vlastní výsledky a také výsledky mezi jednotlivými sportovci.

The screenshot shows a software window titled 'Měření reakční doby sportovce | Porovnání výsledků'. It is divided into two main sections: 'Porovnání sportovce 1' and 'Porovnání sportovce 2'.

**Porovnání sportovce 1:**

- Dropdown menu: st35996
- File selector: BP\_mereni\_st35996\_3\_20150901\_193651.csv
- Table:

Pokus	Meta	Čas	
1	2	857	OK
2	2	781	OK
3	4	993	OK

Výsledky nelze upravovat.

**Summary for Sportovce 1:**

- Celkový čas měření: 00:05:285
- Průměrná reakční doba pro metu 2: 819.0
- Průměrná reakční doba pro metu 4: 993.0
- Minimální čas je lepší o: 175.0 ms
- Maximální čas je lepší o: 34.0 ms
- Průměrný čas je lepší o: 114.5 ms
- Nejdelší reakce: 993.0
- Nejkratší reakce: 781.0
- Průměrná reakční doba: 877.0

**Porovnání sportovce 2:**

- Dropdown menu: st35996
- File selector: BP\_mereni\_st35996\_3\_20150901\_173147.csv
- Table:

Pokus	Meta	Čas	
1	1	1027	OK
2	3	956	OK
3	0	0	Vypršel čas

**Summary for Sportovce 2:**

- Celkový čas měření: 00:06:957
- Průměrná reakční doba pro metu 1: 1027.0
- Průměrná reakční doba pro metu 3: 956.0
- Minimální čas je horší o: 175.0
- Maximální čas je horší o: 34.0
- Průměrný čas je horší o: 114.5
- Nejdelší reakce: 1027.0
- Nejkratší reakce: 956.0
- Průměrná reakční doba: 991.5

At the bottom, there are status indicators: a green checkmark for 'Měření v pořádku' and a red X for 'Neplatné měření'. On the right, there are buttons 'Načíst výsledky' and 'Zavřít'.

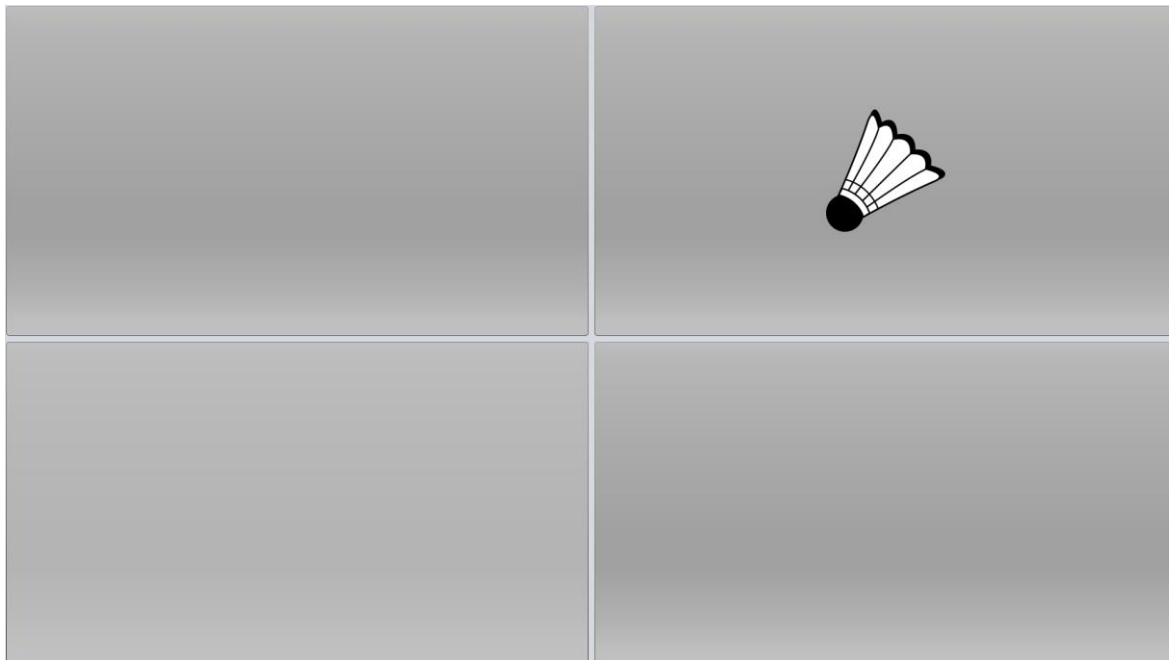
Obrázek 12 – porovnání výsledků

#### 4.2.9 Okno pro sportovce

Okno se otevře na celou obrazovku, obsahující čtyři hlavní tlačítka, která znázorňují mety, a tlačítko pro start. Po skončení měření se toto tlačítko změní na tlačítko pro zavření tohoto okna. Mety jsou znázorněny tlačítky z toho důvodu, aby se aplikace mohla vyzkoušet v offline režimu. V režimu offline zastává funkci sportovce uživatel, aplikace se chová stejně. Měření lze spustit dvěma způsoby:

- mezerníkem na klávesnici,
- stisknutím tlačítka Start.

Po spuštění dojde k zahájení pokusu. Měření se řídí parametry, které jsou nastaveny v předvolbách – viz kapitola 4.2.5. Po spuštění se dolní lišta schová pro lepší soustředění sportovce.



**Obrázek 13 – okno pro sportovce**

První fáze je příprava. Vygeneruje se náhodný čas v rozmezí, které je nastaveno nebo pevně dáno od 2 vteřin do 3 vteřin. Po uplynutí přípravného času se zobrazí na vybrané metě obrázek badmintonového míčku – označující metu ke sešlápnutí (stisknutí). A spustí se odpočítávání času na sešlápnutí (stisknutí) mety - timeout. Mohou nastat tři různé akce.

- 1. akce – sportovec sešlápně zobrazenou metu před vypršením timeoutu.  
(v režimu offline uživatel stiskne zobrazenou metu před vypršením timeoutu)
- 2. akce – sportovec sešlápně špatnou metu před vypršením timeoutu.  
(v režimu offline uživatel stiskne špatnou metu před vypršením timeoutu)
- 3. akce – sportovec nestihne sešlápnout žádnou metu před vypršením timeoutu.  
(v režimu offline uživatel nestihne stisknout žádnou metu před vypršením timeoutu)

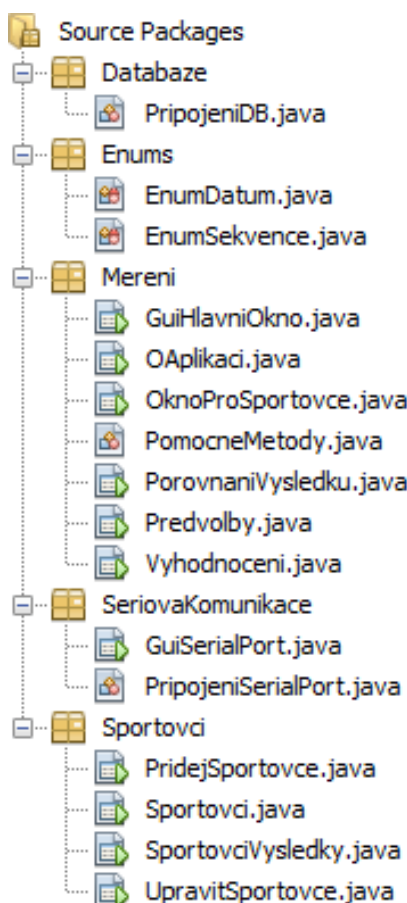
Jedině akce 1 je počítána jako platný pokus. Dojde-li k akci 2 nebo 3, je pokus neplatný. Neplatné pokusy se nezapočítávají do statistiky.

Po sešlápnutí (stisknutí) mety doběhne timeout a sportovec má další čas na přípravu. Čím rychleji sešlápně (stiskne) metu, tím více bude mít času na přípravu. Po každém pokusu se vrací sportovec na startovací značku. Aplikace tento stav nekontroluje.

Až sportovec absolvuje všechny pokusy, bude na obrazovce informován, že měření je dokončeno a v jakém čase. Okno pro sportovce je nutné zavřít stisknutím tlačítka Zavřít.

### 4.3 Implementace aplikace

Aplikace je rozvržena do pěti balíčků. Balíček „SeriovaKomunikace“ obsahuje dvě třídy, které zajišťují připojení zařízení. Třída „GuiSerialPort“ je spouštěcí okno celé aplikace. Balíček „Mereni“ obsahuje třídu „PomocneMetody“. Z této třídy se metody využívají skrz celou aplikaci. Dále se zde nachází třída „GuiHlavniOkno“, toto okno uživatel bude používat po celou dobu spuštěné aplikace. Poté je zde třída „OknoProSportovce“. To je okno, kde se zobrazuje meta, kterou sportovec má sešlápnout. A dále třídy „Predvolby“, „Vyhodnoceni“, „PorovnaniVysledku“ a „OAplikaci“. Další balíček má název „Sportovci“ a obsahuje grafické třídy pro správu sportovce „PridejSportovce“, „Sportovci“, „SportovciVysledky“ a „UpravitSportovce“. Data sportovců jsou uchovávány v databázi a v těchto třídách se pracuje s databázovým souborem. Balíček „Databaze“ obsahuje třídu „PripojeniDB“, která zajišťuje připojení a vytvoření databáze. A poslední balíček je „Enums“ kde se nachází výčetové typy. Výčetový typ „EnumDatum“ uchovává textové položky pro určení data. Výčetový typ „EnumSekvence“ uchovává textové položky sekvencí, které jsou použity jako proměnné v přepínači (switchi).



Obrázek 14 – struktura balíčků aplikace

### 4.3.1 Balíček sériová komunikace

Pro sériovou komunikaci je použita knihovna RXTX. Pro nastavení připojení je potřeba znát správné parametry zařízení:

- přenosovou rychlost,
- počet start-bitů,
- počet stop-bitů,
- nastavení parity.

```
serialPort = (SerialPort) commPort;  
serialPort.setSerialPortParams(  
    115200,  
    SerialPort.DATABITS_8,  
    SerialPort.STOPBITS_1,  
    SerialPort.PARITY_NONE);  
setPripojeno(true);
```

**Zdrojový kód 7 – ukázka nastavení připojení**

Třída pro sériovou komunikaci obsahuje několik metod. Pro zobrazení seznamu dostupných COM portů slouží metoda `void hledaniPortu()`. Připojení a odpojení zařízení mají na starost metody `void pripojit()` a `void odpojit()`. Příjem dat ze zařízení zajišťuje metoda `void serialEvent(SerialPortEvent evt)`. V metodě `int zjistMetu(int vysledek)` se na základě přijímaných dat rozhodne, jaká meta je sešlápnuta. Události jsou ošetřeny zachytáváním výjimek a chybové hlášení je vypisováno do `JLabelu` v grafickém rozhraní „GuiSerialPort“.

### 4.3.2 Balíček měření

Třída `PomocneMetody` obsahuje velkou logiku aplikace. Nachází se zde metoda `int getVygenerovanaMeta(EnumSekvence volba)`, která určuje, jaká meta bude vygenerovaná na základě vybrané sekvence v hlavním okně měření. Metoda vrací číslo mety, která má být sešlápnuta. Tato metoda je volána ze třídy `OknoProSportovce` a návratová hodnota je uložena do proměnné `int vygenerovanaMeta`. Pomocí přepínače (switche) je zjištěno, jestli byla sešlápnuta správná meta, či nikoliv. Tato metoda je k nahlédnutí v příloze B. Dále tu jsou metody pro nastavení tabulek `JTable`. Tabulky se nachází v okně předvoleb, v okně porovnání výsledků a také v hlavním okně měření. Například metoda `void rozmerTabulkyPorovnaní(JTable tabulka, int sirkaOkna)` určuje rozměr sloupců tabulek v okně porovnání výsledků. Tabulka obsahuje čtyři sloupce a poslední sloupec je dopočítáván podle velikosti okna. V metodě je nastavováno zarovnání textu na střed řádku v každém sloupci. Metodu si je možné prohlédnout v příloze C. Nechybí tady ani metoda na vyhodnocování statistiky. Metoda `void vyhodnoceniCasu()` zjišťuje platnost pokusu, minimální reakční čas, maximální reakční čas a průměrnou reakční dobu.

Do této třídy mimo jiné také patří metody pro uložení a načtení výsledků – `void ulozitVysledek()`, `void otevritVysledky(String soubor)`.

Ve třídě „GuiHlavniOkno“ se nachází proměnné typu `Timer` a rozhraní `Runnable`. Rozhraní `Runnable` zajišťuje provádění určitého kódu po spuštění `Timeru`. Na `Timeru` je použita metoda `schedule()`, která očekává tři parametry. Prvním parametrem je `TimerTask()`, ten volá již vytvořené rozhraní `Runnable`. Druhým parametrem je číselná hodnota, za jak dlouho se má určitý kód provést. A třetí parametr očekává periodu v milisekundách. Jsou použity čtyři rozhraní `Runnable`:

- `Runnable runnableAktualniDatum` – pro aktualizování data a času,
- `Runnable runnableCelkovyCas` – pro aktualizaci celkového času měření,
- `Runnable runnableZjistovaniMety` – pro zjišťování sešlápnuté mety,
- `Runnable runnableTimeout` – pro `Timeout`.

Ve třídě „OknoProSportovce“ se v metodě `void startMereni()` nachází již výše zmíněný přepínač (`switch`) na zjištění správné sešlápnuté mety. Mimo jiné se v této metodě nachází proměnné typu `Timer`, kde dochází ke spuštění zjišťování času. Inicializuje se zde:

- `timerDatum` – pro aktualizování data a času,
- `timerCelkovyCas` – pro aktualizaci celkového času měření,
- `timerMety` – pro zjišťování sešlápnuté mety,
- `timerTimeout` – pro `Timeout`.

Ve třídě „Vyhodnoceni“, je metoda `void vyhodnotit()`, kde je volána metoda `vyhodnoceniCasu()` ze třídy `Metody` a je zde nastavována barva písma `JLabelu jLabelVysledekMereni` na základě vyhodnocení výsledku měření.

Ve třídě „Porovnani“ se prochází vytvořené soubory ve složce výsledky, metodou `nacteniSouboru(JComboBox boxMereni, String sportovec)`.

```
void nacteniSouboru(JComboBox comboBoxMereni, String sportovec)
throws IOException {
    comboBoxMereni.removeAllItems();
    Files.walk(Paths.get("./vysledky/" +
sportovec)).forEach(filePath -> {
        if (Files.isRegularFile(filePath)) {
            comboBoxMereni.addItem(filePath.getFileName());
        }
    });
    jButtonNacist.requestFocusInWindow();
}
```

**Zdrojový kód 8 – ukázka metody pro načtení souboru**

Ve třídě „Predvolby“ je mimo jiné metoda pro zachycování uvolnění klávesy na `JTextField`, kde se kontroluje, jestli je správně zadaná hodnota. Pokud není, tak text zčervená a zobrazí se informační dialog `JOptionPane.showMessageDialog()`. V příloze D je možné si prohlédnout kód tlačítka pro uložení nastavení a v příloze E se nachází kód pro uložení sekvence.

### 4.3.3 Balíček sportovci

Třída „Sportovci“ obsahuje metodu `void nactiSportovce()` pro načtení sportovců z databáze.

```
private void nactiSportovce() throws SQLException {
    rs = st.executeQuery("SELECT ID, NetID, OsobniCislo, Jmeno,
    Prijmeni, RokNarozeni, Pohlavi, TypStudia, Rocnik, Fakulta,
    Vyska, Vaha, Sport, Vysledek FROM Sportovec ORDER By NetID");
    String[] sloupce = {"ID", "Net ID", "Osobní číslo", "Jméno",
    "Příjmení", "Rok narození", "Pohlaví", "Typ studia", "Ročník",
    "Fakulta", "Výška", "Váha", "Sport", "Výsledek"};
    DefaultTableModel model = new DefaultTableModel(sloupce, 0);
    dataTable.setModel(model);
    while (rs.next()) {
        Vector<Object> sloupec = new Vector<>();
        sloupec.add(rs.getInt("ID"));
        sloupec.add(rs.getString("NetID"));
        sloupec.add(rs.getString("OsobniCislo"));
        sloupec.add(rs.getString("Jmeno"));
        sloupec.add(rs.getString("Prijmeni"));
        sloupec.add(rs.getInt("RokNarozeni"));
        sloupec.add(rs.getString("Pohlavi"));
        sloupec.add(rs.getString("TypStudia"));
        sloupec.add(rs.getInt("Rocnik"));
        sloupec.add(rs.getString("Fakulta"));
        sloupec.add(rs.getInt("Vyska"));
        sloupec.add(rs.getFloat("Vaha"));
        sloupec.add(rs.getString("Sport"));
        sloupec.add(rs.getString("Vysledek"));
        model.addRow(sloupec);
    }
}
```

**Zdrojový kód 9 – ukázka metody pro načtení sportovce**

V této třídě se také nachází metoda `void rozmerTabulky()` pro nastavení rozměru sloupců tabulky.

Ve třídě „PridejSportovce“ dochází k vložení nového sportovce pomocí dotazu `INSERT INTO Sportovec (NetID, OsobniCislo, Jmeno, Prijmeni, RokNarozeni, Pohlavi, TypStudia, Rocnik, Fakulta, Vyska, Vaha, Sport) VALUES (...)` kde jsou převzaty hodnoty z grafického prostředí.

Třída „UpravitSportovce“ je podobná jako třída „PridejSportovce“. Nepoužívá se zde INSERT, ale UPDATE.

```
conn.setAutoCommit(false);

stmt = conn.createStatement();

String netID = tfNetID.getText();
String osobniCislo = tfOsobniCislo.getText();
String jmeno = tfJmeno.getText();
String prijmeni = tfPrijmeni.getText();
int rokNarozeni = Integer.parseInt(tfRokNarozeni.getText());
String pohlavi = cbPohlavi.getSelectedItem().toString();
String typStudia = cbTypStudia.getSelectedItem().toString();
int rocnik =
Integer.parseInt(cbRocnik.getSelectedItem().toString());
String fakulta = cbFakulta.getSelectedItem().toString();
float vyska = Float.parseFloat(tfVyska.getText());
float vaha = Float.parseFloat(tfVaha.getText());
String sport = tfSport.getText();

String sql = "UPDATE Sportovec SET "
    + "NetID = '" + netID + "', "
    + "OsobniCislo = '" + osobniCislo + "', "
    + "Jmeno = '" + jmeno + "', "
    + "Prijmeni = '" + prijmeni + "', "
    + "RokNarozeni = " + rokNarozeni + ", "
    + "Pohlavi = '" + pohlavi + "', "
    + "TypStudia = '" + typStudia + "', "
    + "Rocnik = " + rocnik + ", "
    + "Fakulta = '" + fakulta + "', "
    + "Vyska = " + vyska + ", "
    + "Vaha = " + vaha + ", "
    + "Sport = '" + sport + "' WHERE id LIKE " + id;

stmt.executeUpdate(sql);
conn.commit();
```

**Zdrojový kód 10 – ukázka metody pro aktualizaci údajů sportovce**

Ve třídě „SportovciVysledky“ je metoda `void nactiVysledky()`. V této metodě se pomocí dotazu `SELECT` načítají do tabulky hodnoty týkající se sportovce a jeho výsledků.

```
private void nactiVysledky() throws SQLException {
    Statement st = conn.createStatement();
    ResultSet rs = null;

    rs = st.executeQuery("SELECT ID, Datum, Cil, Vysledek FROM
Vysledky WHERE NetID LIKE '" + netID + "'");
    String[] sloupce = {"ID", "Datum", "Cil", "Výsledek"};

    DefaultTableModel model = new DefaultTableModel
(sloupce, 0);

    dataTable.setModel(model);

    Calendar cal = Calendar.getInstance();
    TimeZone tz = TimeZone.getTimeZone("+2:00");
    cal.setTimeZone(tz);

    while (rs.next()) {
        Vector<Object> sloupec = new Vector<>();
        sloupec.add(rs.getInt("ID"));
        sloupec.add(rs.getString("Datum"));
        sloupec.add(rs.getInt("Cil"));
        sloupec.add(rs.getString("Vysledek"));

        model.addRow(sloupec);
    }
}
```

**Zdrojový kód 11 – ukázka metody pro načtení výsledků**

Dále v této metodě je načítání seznamu počtu pokusů do `JComboBoxu`.

```
jComboBoxCil.removeAllItems();
rs = st.executeQuery("SELECT DISTINCT Cil FROM Vysledky
WHERE NetID LIKE '" + netID + "' ORDER BY Cil");
while (rs.next()) {
    jComboBoxCil.addItem(rs.getString("Cil"));
}
}
```

**Zdrojový kód 12 – ukázka kódu pro načtení položek do ComboBoxu**



## 4.4 Statistiky

Aplikace zaznamenává a vyhodnocuje následující statistické údaje. Vyhodnocují se pouze platné pokusy – tzn. pokud je v měření neplatný pokus, tak se pro výpočet vynechá. Všechny časové údaje jsou udávány v milisekundách.

### 4.4.1 Platné měření

Prvním vyhodnocením je platnost měření. Tato problematika je trochu nastíněna v kapitole 4.2.9. Aplikace zná dva stavy: měření v pořádku a neplatné měření.

Aby aplikace vyhodnotila **měření v pořádku**, tak sportovec musí ve stanoveném čase (timeout) sešlápnout správnou metu. Jestliže sportovec nestihne sešlápnout metu ve stanoveném čase či sešlápně ve stanoveném čase chybnou metu, výsledek měření se vyhodnotí jako **neplatné měření**.

### 4.4.2 Nejdelší reakce

Nejdelší reakce je zjišťována ze všech platných pokusů v měření pomocí postupného porovnávání hodnot.

```
long max = poleCasu[0];
for (int i = 0; i < poleCasu.length; i++) {
    if (polePlatnosti[i] == true && poleCasu[i] > max) {
        max = poleCasu[i];
    }
}
```

**Zdrojový kód 13 – zjištění nejdelší reakce**

### 4.4.3 Nejkratší reakce

Nejkratší reakce je zjišťována ze všech platných pokusů v měření pomocí postupného porovnávání hodnot.

```
long min = poleCasu[0];
for (int i = 0; i < poleCasu.length; i++) {
    if (polePlatnosti[i] == true && poleCasu[i] < min) {
        min = poleCasu[i];
    }
}
```

**Zdrojový kód 14 – zjištění nejkratší reakce**

#### 4.4.4 Průměrná reakční doba

Průměrná reakční doba ( $\bar{x}$ ) je zjišťována ze všech platných pokusů v měření, následujícím vztahem.

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

$x_i$  – platné pokusy

$n$  – součet všech platných pokusů

```
long prumerCelkem = 0;
for (int i = 0; i < poleCasu.length; i++) {
    if (polePlatnosti[i] == true) {
        prumerCelkem += poleCasu[i];
    }
}
prumerCelkem = prumerCelkem / poleCasu.length;
```

**Zdrojový kód 15 – zjištění průměrné reakční doby**

## Závěr

Hlavním cílem bakalářské práce bylo vytvoření snadno použitelné aplikace, která umožní nastavení základních parametrů pro měření reakční doby sportovce a tuto reakční dobu změřit. K hlavnímu cíli patří také vyhodnocení statistik měření. Aplikace v tomto duchu byla vyvíjena. Postupem času jsem do aplikace přidal i seznam sportovců, ve kterém jsou uloženy základní údaje sportovce. Tento seznam je vytvořen pomocí databázové tabulky za použití SQL.

Práce se skládá z teoretické a praktické části. V teoretické části je popsán programovací jazyk Java, databáze SQLite, sériová komunikace a nativní rozhraní Javy. Praktická část se zabývá vývojem a implementací aplikace.

Popis programovacího jazyka Java je popsán v základních informacích. Snažil jsem se vystihnout podstatné věci a především ty věci, které ve své práci implementuji.

Programovací jazyk Java patří mezi nejpoužívanější programovací jazyky na světě a vybral jsem si tento jazyk kvůli své jednoduchosti a přenositelnosti. Ovšem neměl jsem možnost aplikaci vyzkoušet na jiném operačním systému než na Windows. Pro správu databáze jsem vybral SQLite díky své jednoduchosti a především licenčním podmínkám, která je volně šiřitelná a použitelná zdarma. SQLite postačuje se svojí využitelností i pro mnohem složitější a rozsáhlejší aplikace. Ve své práci s databází pracuji jen okrajově a především ji využívám pro usnadnění a efektivnější práci nejen pro správu sportovců. Pro sériovou komunikaci byla zvolena nativní knihovna RXTX, jedná se o alternativní knihovnu pro sériovou komunikaci, která je založena na specifikaci oficiální knihovny pro sériovou komunikaci, kterou vyvinula společnost Sun.

Praktická část se zabývá implementací aplikace. Popisuje návrh, rozvržení a samotnou implementaci. Aplikaci jsem se snažil rozvrhnout tak, aby byla jednoduchá na ovládání, na pochopení a obsahovala všechny důležité prvky. Důležitým faktorem pro vývoj bylo především dosáhnout největší komfort pro sportovce, aby se sportovec mohl maximálně soustředit na svůj výkon. Bohužel jsem měl k dispozici pouze testovací zařízení, takže jsem nemohl provést ostrý test se samotným sportovcem. Testování zastával můj vedoucí práce, Ing. Martin Lauterbach, který měl podmětné připomínky a návrhy, které mi pomohly v dalším vývoji.

Během vývoje aplikace, jako snad u každého vývoje, se vyskytovaly problémy, které zpomalovaly vývoj, ale všechny problémy se podařilo vyřešit. Aplikace je ve finální verzi plně použitelná a splňuje všechny stanovené cíle. Úspěšně se také podařilo vytvořit odpovídající adresářovou strukturu aplikace, která odděluje aplikační data od uživatelských dat. Především se podařilo naprogramovat aplikaci tak, aby sama poznala, jakou verzi operačního systému Windows uživatel používá a podle toho si sama zvolí správnou verzi knihovny. Díky tomu jsem nebyl nucen aplikaci vydávat ve dvou verzích, jak ve verzi pro 32-bit operační systém a pro 64-bit operační systém.

Aplikace je připravena pro další vývoj, úpravy a rozšíření, na základě požadavků od jednotlivých uživatelů. Především o podrobnější statistické údaje nebo grafické vyhodnocení v podobě srovnávacích grafů.

Součástí této práce je také příručka uživatele, která je přiložena jako příloha F. Tato příručka podrobně popisuje funkcionalitu aplikace pro snadnější pochopení a také obsahuje důležité informace pro používání aplikace.

## Literatura

- [1] HORTON, Ivor. *Java 5*. Praha: Neocortex, 2005, 1443 s. ISBN 80-863-3012-5.
- [2] HEROUT, Pavel. *Java: grafické uživatelské prostředí a čeština*. 2. vyd. České Budějovice: Kopp, 2007, 316 s. ISBN 978-80-7232-328-9.
- [3] JARVI, Keane. RXTX. RXTX wiki [online]. 2011 [cit. 2013-11-01]. Dostupné z: <http://rxtx.qbang.org>
- [4] Serial Programming/Serial Java. *Wikibooks* [online]. 2015 [cit. 2015-07-02]. Dostupné z: [https://en.wikibooks.org/wiki/Serial\\_Programming/Serial\\_Java](https://en.wikibooks.org/wiki/Serial_Programming/Serial_Java)
- [5] RXTX for Windows. *JLog* [online]. 2011 [cit. 2015-07-02]. Dostupné z: <http://jlog.org/rxtx-win.html>
- [6] LACKO, Ľuboslav. *1001 tipů a triků pro SQL*. Vyd. 1. Brno: Computer Press, 2011, 416 s. ISBN 978-80-251-3010-0
- [7] Úvod do JDBC. ŠEDA, Jan. *Interval.cz* [online]. 2003 [cit. 2015-08-05]. Dostupné z: <https://www.interval.cz/clanky/uvod-do-jdbc/>
- [8] *SQLite: Small. Fast. Reliable. Choose any three.* [online]. 2015 [cit. 2015-08-05]. Dostupné z: <http://www.sqlite.org/>
- [9] SQLite Java Tutorial. *Tutorialspoint: SimplyEasyLearning* [online]. 2015 [cit. 2015-08-05]. Dostupné z: [http://www.tutorialspoint.com/sqlite/sqlite\\_java.htm](http://www.tutorialspoint.com/sqlite/sqlite_java.htm)
- [10] Java Native Interface. 1993. *Java SE Documentation* [online]. 2014 [cit. 2015-08-05]. Dostupné z: <http://docs.oracle.com/javase/7/docs/technotes/guides/jni/spec/intro.html>

## **Příloha A – CD s aplikací a zdrojovými kódy**

Obsah CD je následující:

- adresář `aplikace` – spustitelná aplikace,
- adresář `textova_cast` – textová část bakalářské práce,
- adresář `zdrojove_kody` – zdrojové kódy aplikace.

## Příloha B – ukázka metody getVygenerovanaMeta třídy PomocneMetody

```
public static int getVygenerovanaMeta(EnumSekvence volba) {
    int cislo;
    switch (volba) {
        case NAHODNE:
            cislo = (int) (Math.random() * 4 + 1);
            OknoProSportovce.vygenerovanaMeta = cislo;
            switch (cislo) {
                case 1:
                    vygenerovanaMetaJedna();
                    break;
                case 2:
                    vygenerovanaMetaDva();
                    break;
                case 3:
                    vygenerovanaMetaTri();
                    break;
                case 4:
                    vygenerovanaMetaCtyri();
                    break;
                default:
                    return 0;
            }
            return cislo;

        case UZIVATELSKY_VYBIRANE:
            Object[] vyber = {"1", "2", "3", "4"};
            cislo = (JOptionPane.showOptionDialog(null,
                "Vyberte metu",
                "Výběr mety",
                JOptionPane.YES_NO_CANCEL_OPTION,
                JOptionPane.DEFAULT_OPTION,
                null,
                vyber,
                vyber[0])) + 1;

            switch (cislo) {
                case 1:
                    vygenerovanaMetaJedna();
                    break;
                case 2:
                    vygenerovanaMetaDva();
                    break;
                case 3:
                    vygenerovanaMetaTri();
                    break;
                case 4:
                    vygenerovanaMetaCtyri();
                    break;
                default:
                    return 0;
            }
    }
}
```

```

        return cislo;

        case NADEFINOVANA_SEKVENCE:
            cislo =
Predvolby.poleNastaveniMet[pocitadloUzivNastavenaSekvence];
            pocitadloUzivNastavenaSekvence++;

            switch (cislo) {
                case 1:
                    vygenerovanaMetaJedna();
                    break;
                case 2:
                    vygenerovanaMetaDva();
                    break;
                case 3:
                    vygenerovanaMetaTri();
                    break;
                case 4:
                    vygenerovanaMetaCtyri();
                    break;
                default:
                    return 0;
            }
            return cislo;
        }
        return 0;
    }
}

```



## Příloha C – ukázka metody rozmerTabulkyPorovnani třídy PomocneMetody

```
public static void rozmerTabulkyPorovnani(JTable tabulka, int
sirkaOkna) {
    if (vymazatTabulku) {
        String[] sloupce = {"Pokus", "Meta", "Čas", ""};
        tabulka.setModel(new DefaultTableModel(sloupce, 0));
    }

    tabulka.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);

    //šířka
    int zbytek = 36;
    int sloupec1 = 60; //pokus
    int sloupec2 = 60; //meta
    int sloupec3 = 100; //čas
    int sloupec4 = ((sirkaOkna - sloupec1 - sloupec2 -
sloupec3) - zbytek); //text

    tabulka.getColumnModel().getColumn(0).setPreferredWidth(sloupec1);
    tabulka.getColumnModel().getColumn(1).setPreferredWidth(sloupec2);
    tabulka.getColumnModel().getColumn(2).setPreferredWidth(sloupec3);
    tabulka.getColumnModel().getColumn(3).setPreferredWidth(sloupec4);

    //zarovnání textu řádku doprostřed
    DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
    centerRenderer.setHorizontalAlignment(JLabel.CENTER);

    tabulka.getColumnModel().getColumn(0).setCellRenderer(centerRender
er);

    tabulka.getColumnModel().getColumn(1).setCellRenderer(centerRender
er);

    tabulka.getColumnModel().getColumn(2).setCellRenderer(centerRender
er);

    tabulka.getColumnModel().getColumn(3).setCellRenderer(centerRender
er);
```

## Příloha D – ukázka kódu tlačítka pro uložení nastavení ze třídy Predvolby

```
private void
jButtonUlozitNastaveniActionPerformed(java.awt.event.ActionEvent
evt) {
    try {
        if (jRadioButtonRandom.isSelected()) { //vychozi
nastavení
            //Nastavení timeoutut
            GuiHlavniOkno.timeout = 5000;
GuiHlavniOkno.jLabelTimeout.setText(String.valueOf(GuiHlavniOkno.t
imeout));

            //Nastavení počet pokusů
            GuiHlavniOkno.pocetPokusu = 5;
GuiHlavniOkno.jLabelCilZbyvaHodnota.setText(String.valueOf(GuiHlav
niOkno.pocetPokusu));

            //Nastavení přípravného času
            OknoProSportovce.dolniMezPripravy = 2000;
            OknoProSportovce.horniMezPripravy = 3000;

            nastaveniTextFieldu();
        } else if (jRadioButtonNastavit.isSelected()) {
//nastavené nastavení uživatelem
            //Nastavení timeoutut
            GuiHlavniOkno.timeout =
Long.parseLong(jTextFieldTimeout.getText());
GuiHlavniOkno.jLabelTimeout.setText(String.valueOf(GuiHlavniOkno.t
imeout));

            //Nastavení počet pokusů
            GuiHlavniOkno.pocetPokusu =
Integer.parseInt(jTextFieldPocetPokusu.getText());
GuiHlavniOkno.jLabelCilZbyvaHodnota.setText(String.valueOf(GuiHlav
niOkno.pocetPokusu));

            //Nastavení přípravného času
            OknoProSportovce.dolniMezPripravy =
Integer.parseInt(jTextFieldDolniMez.getText());
            OknoProSportovce.horniMezPripravy =
Integer.parseInt(jTextFieldHorniMez.getText());
        }
        JOptionPane.showMessageDialog(this, "Nastavení bylo
uloženo.");
    } catch (NullPointerException ex) {
        JOptionPane.showMessageDialog(rootPane, "Chybné
zadání.\nPřečtěte si nápovědu u každé položky při najetí myši.");
    }
}
}
```

## Příloha E – ukázka kódu tlačítka pro uložení sekvence ze třídy Predvolby

```
private void
jButtonUlozitSekvenciActionPerformed(java.awt.event.ActionEvent
evt) {
    try {
        //vytažení dat z JTable
        for (int i = 0; i < GuiHlavniOkno.pocetPokusu; i++) {
            poleNastaveniMet[i] = Integer.parseInt((String)
jTableUzivSekvence.getValueAt(i, 1));
            poleNastaveniCasu[i] = Long.parseLong((String)
jTableUzivSekvence.getValueAt(i, 2));
        }

        for (int i = 0; i < GuiHlavniOkno.pocetPokusu; i++) {
            System.out.println("meta: " +
poleNastaveniMet[i]);
            System.out.println("čas: " +
poleNastaveniCasu[i]);
        }

        OknoProSportovce.uzivatelskyTimeout = true;

GuiHlavniOkno.jRadioButtonMenuItemSNadefinovanoUzivatelem.setSelected(true);

        JOptionPane.showMessageDialog(this, "Sekvence byla
uložena.");
    } catch (ArrayIndexOutOfBoundsException |
NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Je nutné vyplnit
všechny údaje a potvrdit Enterem.");
    }
}
```

# **Aplikace pro měření reakční doby sportovce**

## **Příručka uživatele**

## Obsah

<b>1</b>	<b>Instalace.....</b>	<b>53</b>
1.1	Minimální požadavky.....	53
1.2	Instalace aplikace.....	53
<b>2</b>	<b>Základní informace .....</b>	<b>53</b>
<b>3</b>	<b>Výchozí nastavení aplikace .....</b>	<b>54</b>
<b>4</b>	<b>Spuštění aplikace .....</b>	<b>54</b>
4.1	Připojení zařízení.....	54
4.2	Okno měření .....	55
4.2.1	Nabídka Nastavení – Sportovci .....	56
4.2.2	Nabídka Nastavení – Předvolby .....	56
4.2.3	Nabídka Sekvence .....	57
4.2.4	Nabídka Výsledky – Vyhodnocení.....	57
4.2.5	Nabídka Výsledky – Porovnání.....	58
4.2.6	Nabídka Nastavení.....	58
4.3	Okno pro sportovce .....	59

# 1 Instalace

Aplikace je vytvořena pro operační systém Microsoft Windows. Funkčnost na jiném OS není zaručena ani otestována.

## 1.1 Minimální požadavky

Než se pustíte do instalace, ujistěte se, zda Váš počítač splňuje minimální požadavky pro správné fungování aplikace.

**Operační systém:** Microsoft Windows 7

**Monitor:** rozlišení 1280x800

**Konektivita:** USB port

**Software:** Java Version 8 Update 60

## 1.2 Instalace aplikace

Aplikace je dodávána v komprimovaném souboru metodou ZIP. Aplikace se neinstaluje, jen se komprimovaný soubor extrahuje (rozbalí). Rozbalení se provede přes kontextovou nabídku vyvolanou pravým tlačítkem myši na komprimovaný soubor a vybere se volba rozbalit vše. Po rozbalení se vytvoří složka s názvem „ReakcniDoba“ s níže uvedenou adresářovou strukturou, která se musí zachovat. Celou rozbalenou složku lze přesunout na libovolné umístění. Soubory a složky uvnitř rozbalené složky se nesmějí přesouvat ani přejmenovávat.

Ujistěte se, že máte nainstalovanou aktuální verzi Javy, tím předejdete případným problémům.

# 2 Základní informace

- Nejdůležitější prvky v aplikaci jsou opatřeny doprovodným textem – při najetí myši na prvek se zobrazí krátký popis co ten prvek znamená/vykonává.
- Dbejte prosím na správně zadané parametry.
- Soubor s uloženými výsledky měření má následující podobu:

**BP\_mereni\_st35996\_3\_20150830\_110118.csv**

**Bakalářská práce\_měření\_Net ID\_počet pokusů\_rrrrmdd\_hhmmss.csv**

**struktura názvu souboru**

### 3 Výchozí nastavení aplikace

Zde je popsáno, jak jsou nastaveny parametry ve výchozím stavu.

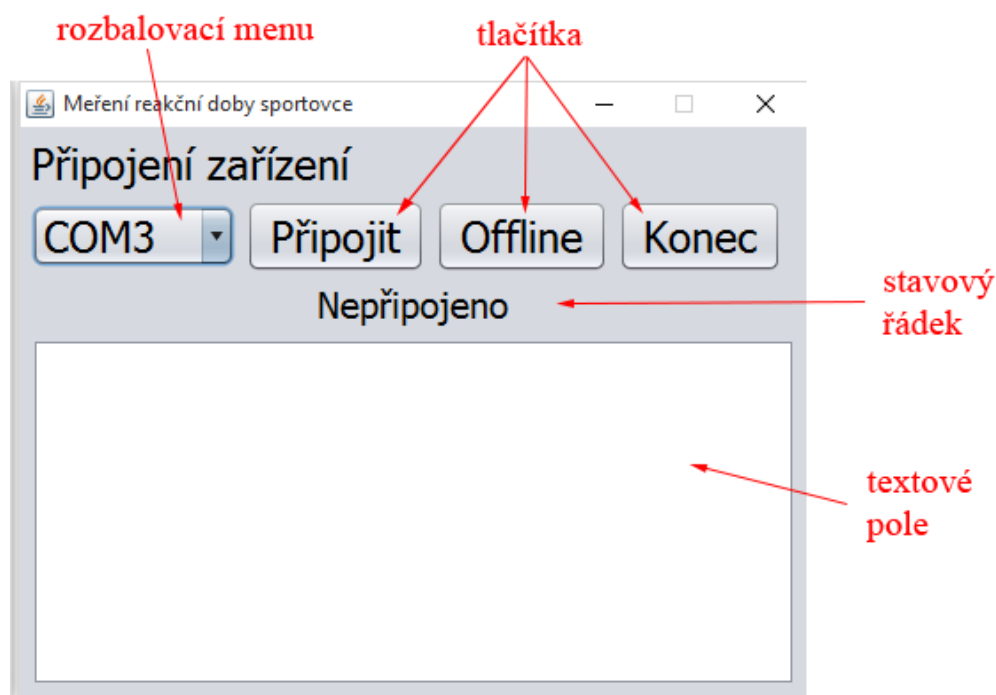
- Timeout – 5 000 ms,
- Počet pokusů – 5,
- Dolní mez přípravy – 2 000 ms,
- Horní mez přípravy – 3 000 ms.

### 4 Spuštění aplikace

Aplikace se spouští souborem „ReakniDoba.jar“.

#### 4.1 Připojení zařízení

Po spuštění aplikace se zobrazí okno pro připojení zařízení. Okno informuje o aktuálním stavu připojeného zařízení. Po kliknutí na rozbalovací seznam dojde k aktualizaci COM portů. Pokud zařízení není připojené, lze vyzkoušet offline režim.



okno připojení zařízení

**Upozornění:** Dbejte prosím zvýšené opatrnosti při zapojeném zařízení. Pokud by došlo k odpojení zařízení při spuštěném měření, měření by nebylo korektní a muselo by se celé opakovat. Pokud máte k PC připojeno více zařízení, které jsou rozpoznávány na portu COM, doporučuji zařízení pro měření reakční doby připojit až po spuštění aplikace, abyste měli jistotu, že vybraný COM port je správný.

## 4.2 Okno měření

V tomto okně se zobrazují veškeré informace potřebné k měření. Přes nabídkovou lištu lze otevřít další okna aplikace. Některé volby se aktivují až po určité akci – lze zjistit po najetí myši na danou položku.

Měření reakční doby sportovce

Nastavení Sekvence Výsledky Nastavení **nabídková lišta**

Dnes je: 1.9.2015 10:15:35 **aktuální datum**

**00:00:000** **celkový čas měření**

**5000** **timeout**

Pokus	Meta	Čas
<b>tabulka zaznamenaných pokusů</b>		

Počet pokusů: 0 Zbývá pokusů: 5

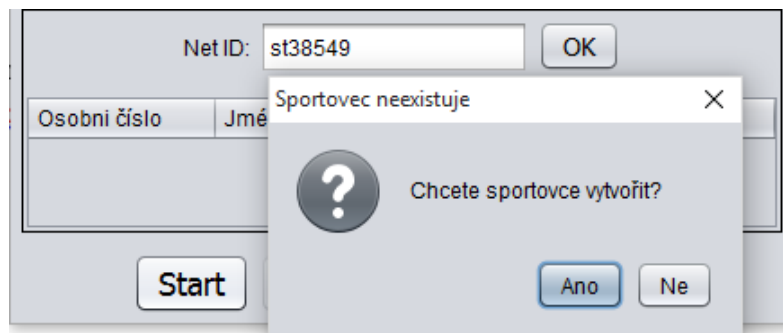
Net ID:

<b>tabulka zadaného sportovce</b>
-----------------------------------

okno měření

Před spuštěním měření se musí nejdříve zadat Net ID sportovce a tlačítkem „OK“ vyhledat v databázi sportovců – pokud sportovec nebude existovat, je nutné sportovce vytvořit.





okno při neexistujícím sportovci

**Upozornění:** Pokud používáte více jak jeden monitor a chcete při měření kontrolovat stav v tomto okně, je nutné toto okno přetáhnout na jiný monitor před stisknutím tlačítka Start. Okno pro sportovce se automaticky otevře na celou obrazovku monitoru, který je nastaven jako hlavní monitor. Tlačítka Start, Reset, Uložit a Konec jsou opatřeny potvrzujícím dialogem, zda určitou funkcionalitu chcete opravdu provést.

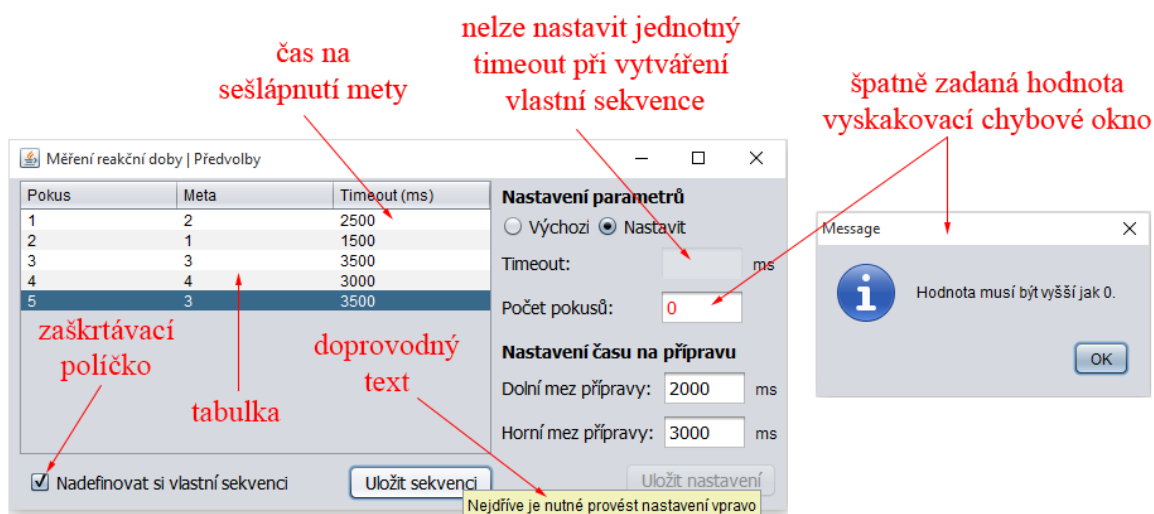
#### 4.2.1 Nabídka Nastavení – Sportovci

V okně sportovci se dá přidat nový sportovec, upravit nebo smazat existující sportovec. U sportovců si lze prohlédnout dosažené výsledky.

**Upozornění:** Při smazání sportovce nedochází k smazání zaznamenaných měření, z důvodu nechtěného smazání. Výsledky je poté nutné z počítače smazat ručně. O evidenci se stará databázový soubor ReakcniDoba.db – nedoporučuje se tento soubor upravovat jiným způsobem než přímo v aplikaci.

#### 4.2.2 Nabídka Nastavení – Předvolby

V předvolbách se nastavují parametry měření a definuje vlastní sekvence. Po vytvoření a uložení vlastní sekvence dochází automaticky k vybrání této sekvence pro měření.



okno předvolby

**Důležité:** Je třeba si pořádně promyslet sekvenci. Do již vytvořené sekvence nejdu vkládat a odebírat pokusy, lze jen editovat již předem daný počet pokusů. Nastavovaný čas je v milisekundách.

#### 4.2.3 Nabídka Sekvence

Nabídka Sekvence nabízí na výběr nadefinované sekvence.

- Náhodné – mety se budou zobrazovat náhodně,
- Uživatelsky vybírané – před každým zobrazením mety se aplikace dotáže jaká meta má být zobrazena, metu si zvolíte v zobrazeném vstupním dialogu,
- Postupně – mety se budou zobrazovat postupně v pořadí: 1,2,3,4,1 atd.,
- Sudé->Liché – mety se budou zobrazovat v pořadí: 2,4,1,3,2 atd.,
- Liché->Sudé – mety se budou zobrazovat v pořadí: 1,3,2,4,1 atd.,
- Nadefinovaná sekvence – tato volba je z této nabídky neaktivní, lze aktivovat jen nastavením vlastní sekvence v předvolbách, viz kapitola 4.2.2.

#### 4.2.4 Nabídka Výsledky – Vyhodnocení

Okno, v kterém je zobrazeno vyhodnocení aktuálního skončeného měření. Položka ve výchozím nastavení je neaktivní, po skončení měření se zaktivní a lze si prohlédnout vyhodnocení.

Naleznete zde informaci, zda proběhlo měření v pořádku. Dozvíte se celkový čas měření, nejdelší reakci, nejkratší reakci a průměrnou reakční dobu.

Pokus	Meta	Čas	
1	1	1027	OK
2	3	956	OK
3	0	0	Vypršel čas

Měření reakční doby sportovce | Vyhodnocení

### Vyhodnocení pro sportovce st35996

**Neplatné měření**

Celkový čas měření: 00:06:957  
Nejdelší reakce: 1027  
Nejkratší reakce: 956  
Průměrná reakční doba: 661

vyhodnocení aktuálního měření

#### 4.2.5 Nabídka Výsledky – Porovnání

V porovnání se porovnávají mezi sebou dva výsledky měření.

Na výběr jsou všichni sportovci, kteří jsou zaneseni v databázi. Takže na výběr bude i sportovec, který ještě neabsolvoval žádné měření. Sportovec, který již měření absolvoval, tak po jeho výběru je umožněno vybrat konkrétní měření. Poté výběr potvrdíte tlačítkem „Načíst výsledky“, nebo stiskem mezerníku. V tabulce se zobrazí výsledky a pod ní souhrnné informace. Lze porovnávat své vlastní výsledky a také výsledky mezi jednotlivými sportovci.

**Porovnání sportovce 1**

st35996

BP\_mereni\_st35996\_3\_20150901\_193651.csv

Pokus	Meta	Čas	
1	2	857	OK
2	2	781	OK
3	4	993	OK

Výsledky nelze upravovat.

Celkový čas měření: 00:05:285

Průměrná reakční doba pro metu 2: 819.0  
Průměrná reakční doba pro metu 4: 993.0

Minimální čas je lepší o: 175.0 ms  
Maximální čas je lepší o: 34.0 ms  
Průměrný čas je lepší o: 114.5 ms

Nejdelší reakce: 993.0  
Nejkratší reakce: 781.0  
Průměrná reakční doba: 877.0

Text nelze upravovat.

**Porovnání sportovce 2**

st35996

BP\_mereni\_st35996\_3\_20150901\_173147.csv

Pokus	Meta	Čas	
1	1	1027	OK
2	3	956	OK
3	0	0	Vypršel čas

Celkový čas měření: 00:06:957

Průměrná reakční doba pro metu 1: 1027.0  
Průměrná reakční doba pro metu 3: 956.0

Minimální čas je horší o: 175.0  
Maximální čas je horší o: 34.0  
Průměrný čas je horší o: 114.5

Nejdelší reakce: 1027.0  
Nejkratší reakce: 956.0  
Průměrná reakční doba: 991.5

✓ Měření v pořádku      ✗ Neplatné měření      Načíst výsledky      Zavřít

porovnání výsledků

#### 4.2.6 Nabídka Nastavení

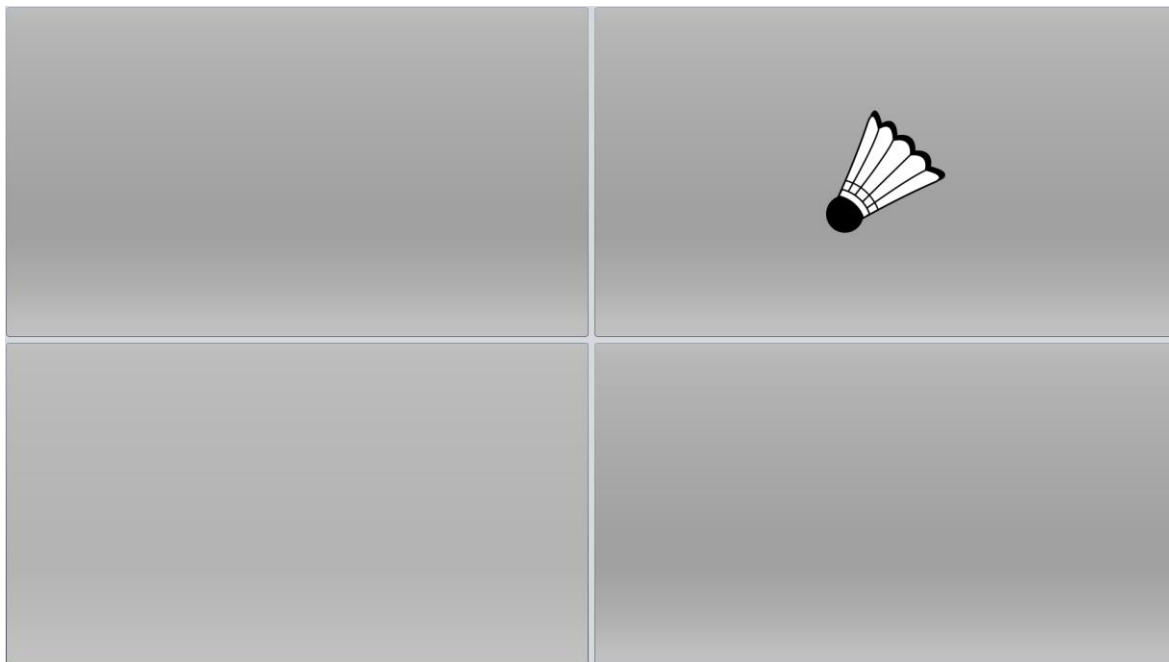
Nabídka Nastavení obsahuje jednu položku – O aplikaci. Kde je uvedena verze a autor aplikace.

### 4.3 Okno pro sportovce

Po zadání Net ID sportovce a vyresetování předchozího měření se po stisku tlačítka „Start“ zobrazí na celou obrazovku okno pro sportovce.

Nezbývá než sportovce postavit na start a stiskem klávesnice mezerníku nebo tlačítkem start zahájit měření.

Po spuštění se dolní lišta schová a po skončení měření se opět objeví.



okno pro sportovce