

**Gymnázium, Praha 6, Arabská 14**

**Programování**

**ROČNÍKOVÝ PROJEKT**

**Výuka slovíček**



Duben 2022

Michael Vakula, 1.E

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V ..... dne .....

Michael Vakula .....

## **Anotace:**

Tento ročníkový projekt obsahuje program, sloužící k procvičování a výuce cizích slovíček za pomoci náhodného losování slov z už daného seznamu slovíček. Program má funkci počítání bodů, které uživatel získal a funkci zvýšení šance výskytu daného slovíčka, které bylo zodpovězené nesprávně. Celý program byl napsán v programovacím jazyce Java a pro grafické rozhraní byla využita platforma JavaFX.

## **Abstract:**

This year's project contains a program used to practice and teach foreign words by randomly drawing words from a given list of words. The program has the function of counting the points that the user earned and the function of increasing the chance of occurrence of a given word that was answered incorrectly. The whole program was written in the Java programming language and the JavaFX platform was used for the graphical interface.

## **Zadání:**

Program bude generovat např. 10-20 náhodných slovíček z vybrané lekce a cílem bude je napsat správně anglicky. Za každou správnou odpověď se přičte bod a ke konci se body sečtou. Program si bude pamatovat úspěšnost a nabízet častěji ta slovíčka která mají menší úspěšnost.

# Obsah

|   |    |
|---|----|
| Úvod.....                                 | 6  |
| 1.Vývojové prostředí.....                 | 7  |
| 1.1.Java.....                             | 7  |
| 1.2.JavaFX.....                           | 7  |
| 1.3.IntelliJ.....                         | 7  |
| 2.Struktura funkcí programu v Java.....   | 8  |
| 2.1. Klíčové třídy a metody.....          | 8  |
| 2.2.Zpracování funkcí.....                | 8  |
| 3.Grafické provedení.....                 | 10 |
| 3.1. Popis grafiky a její stylizace ..... | 10 |
| 3.2. Implementace funkcí.....             | 12 |
| Závěr.....                                | 14 |
| Zdroje.....                               | 15 |
| Seznam obrázků.....                       | 16 |

## Úvod

Zvolené téma je program, který umožňuje procvičování už vybraných cizích slovíček ve formě náhodného vybrání slova, které musí uživatel správně přeložit do českého jazyka.

Toto téma bylo návrh na ročníkovou práci od mého učitele na programování, Mgr. Jan Lána, tento návrh jsem si oblíbil a při postupném vytváření programu mi velice sedl. Základní myšlenka programu je obohacena o několik funkcí, které dodávají na obtížnosti a rozmanitosti programu.

Pro vytváření programu jsem využil programovací jazyk Java a vývojové prostředí IntelliJ, které používáme v hodinách předmětu programování, a také se v něm přehledněji pracuje.

## **1. Vývojové prostředí**

Za prostředí, ve kterém jsem vypracoval tento program jsem si zvolil IntelliJ IDEA Community Edition 2021.3.2 s programovacím jazykem Java 17.

## 1.1. Java

Java je programovací jazyk a výpočetní platforma vydaná společností Sun Microsystems v roce 1995. Vyvinula se ze skromných začátků, aby poháněla velkou část dnešního digitálního světa tím, že poskytuje spolehlivou platformu, na které je postaveno mnoho služeb a aplikací.

Java je zdarma ke stažení pro osobní použití. Nejnovější verzi můžete získat na [java.com](https://www.oracle.com/javadownload/) a všechny vývojové sady a další užitečné nástroje najdete na <https://www.oracle.com/javadownload/>. [1]

## 1.2. JavaFX

JavaFX je open source klientská aplikační platforma nové generace pro stolní, mobilní a vestavěné systémy postavená na Javě. Jedná se o společné úsilí mnoha jednotlivců a společností s cílem vytvořit moderní, efektivní a plně funkční sadu nástrojů pro vývoj bohatých klientských aplikací. [2]



Obrázek 1 – logo programovacího jazyka Java

## 1.3. IntelliJ

IntelliJ IDEA je integrované vývojové prostředí (IDE) pro jazyky JVM navržené tak, aby maximalizovalo produktivitu vývojářů. Provádí rutinní a opakující se úkoly za vás tím, že poskytuje chytré dokončování kódu, analýzu statického kódu a umožňuje vám tak soustředit se na světlou stránku vývoje softwaru. [3]

# 2. Struktura funkcí programu v Java

Na začátek je dobré zmínit, že program jsem nejdříve vytvářel pouze v jazyce Java, a až poté, jsem se zaměřil na grafické zpracování, proto jsem se rozhodl věnovat tuto kapitolu pouze pro vytváření programu v Java.

V následujících podkapitolách se tedy budu věnovat důležitým třídám a metodám, které v programu hrají klíčovou roli

## 2.1. Klíčové třídy a metody

Za nejdůležitější třídu v mém programu považuji ArrayList, třída ArrayList je pole prvků s měnitelnou velikostí, prvky zde jde kdykoliv přidávat. Pro vytvoření takového ArrayListu je nejdříve nutné implementovat import: `java.util.ArrayList`. A pro samotné vytvoření je zapotřebí část kódu: `ArrayList<String> nazev = new ArrayList<String>();`. Mezi využívané metody patří metoda `add`, pro přidání prvku, a metoda `get`, která vrátí prvek ze zadané pozice z daného seznamu. [4]

K umožnění komunikace klávesnice s programem jsem použil třídu `Scanner`. Pro její vytvoření si musíme přidat import: `java.util.Scanner`. Dále je za potřebí si vytvořit vlastní scanner a udělat si proměnnou hodnotu, která se bude rovnat vstupu z klávesnice, to provedeme takto: `Scanner sc = new Scanner(System.in); int i = sc.nextInt();`. [5]

Pro vybrání náhodné hodnoty jsem využil třídu `Random`. Pro vytvoření je zapotřebí import: `java.util.Random`. A kód pro vytvoření náhodného generátoru a neznámá která ho bude obsahovat: `Random nazev = new Random();`, `nazev` proměnné = `nazev.nextInt(nejvyšší číslo);`. [6]

## 2.2. Zpracování funkcí

Jako první jsem si vytvořil ArrayList pro česká i německá slovíčka:

```
public void start(Stage stage){  
    //Zde jsou přidány česká slova  
    cesta.add("snídaně");  
    cesta.add("snídat");  
    cesta.add("zelenina");  
    //Přidána německá slova  
    nemcina.add("Frühstück");  
    nemcina.add("frühstücken");  
    nemcina.add("das Gemüse");  
}
```

Obrázek 2 až 3 - Vytvoření ArrayListu

Když české slovíčko je na pozici nula v ArrayListu, prvek s německým překladem daného slova musí být na stejné pozici ve svém ArrayListu.

Dále jsem si vytvořil proměnnou, která nese vstup z klávesnice, proměnnou, která obsahuje náhodné číslo od nuly do velikosti ArrayListu českých slovíček, a proměnnou s počáteční hodnotou nula, která složí pro počítání bodů. Myšlenka je taková, že se vypíše prvek z náhodné pozice v ArrayListu českých slovíček, poté se bude chtít vstup z klávesnice, který



se buď bude rovnat prvku z náhodné pozice ArrayListu německých slovíček, v takovém případě by program napsal “správně”, přičetl jeden bod a vypsals další české slovo, pokud se však nebude rovnat, program napíše “špatně” a dané slovíčko přidá do následující pozice v ArrayListu a to samé udělá s překladem. Tím by se v ArrayListu objevovali dvě stejná slova na různých pozicích a šance, že se dané slovo objeví znovu by se zvýšila.

Pro tento návrh je však velmi důležité, aby české slovo a jeho překlad byly na stejných pozicích v obou ArrayListech.

```
int index = a.nextInt(cesky.size());

for (i = 0; i < pocetSlov; i++) {

    index = a.nextInt(cesky.size());

    System.out.println(cesky.get(index));
    odpoved = skenr.nextLine();
    if (odpoved.equals(nemecky.get(index))) {
        System.out.println("spravne");
        pocetBodu = b++;
    } else {
        System.out.println("spatne");
        cesky.add(cesky.get(index));
        nemecky.add(nemecky.get(index));
    }
}
```

Obrázek 4 – Funkce programu

Pro účel, aby program nevypsals jen jedno slovo a neskončil, jsem použil příkaz for, který opakuje část programu, dokud nebude splněna určitá podmínka. [7] V mém případě jsem si vytvořil proměnnou s počáteční hodnotou nula a přičítal k ní jedničku, dokud proměnná nebude mít stejnou hodnotu, jako proměnná “pocetSlov”.

Ve smyčce for je také obsažena proměnná obsahující náhodné generované číslo a proměnná se vstupem z klávesnice. To z toho důvodu, že chceme pokaždé jiné české slovo a pro něj i jiný vstup z klávesnice.

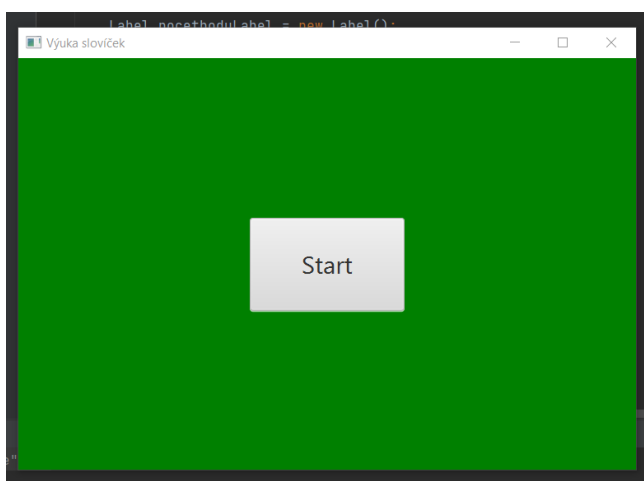
### 3. Grafické provedení

V této kapitole budu popisovat grafické rozhraní programu a jeho stylizaci.

#### 3.1. Popis grafiky a její stylizace

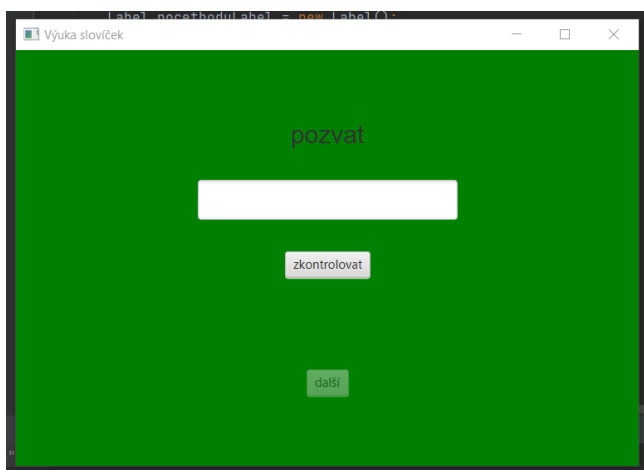
V mém programu se objevují JavaFX prvky, které si teď představíme. Jako první je zde Button, což je ovládací prvek umožňující aplikaci JavaFX provést nějakou akci, když uživatel aplikace klepne na tlačítko. Label je ovládací prvek, který může zobrazit textový nebo obrázkový štítek uvnitř JavaFX GUI. TextField umožňuje uživatelům aplikace JavaFX zadávat text, který pak může aplikace číst, VBox je komponent, který umísťuje všechny jeho prvky do svislého sloupce na sebe. [8]

Můj program obsahuje tři scény. První scéna má sloužit jako startovní panel a obsahuje pouze tlačítko, které vás přenese na druhou scénu.



Obrázek 5 - První scéna

Druhá scéna se dá považovat za hlavní, proto je i nejvíce obsáhlá. Jako první je v ní Label, do kterého se vypíše náhodné české slovo, poté je zde TextField, kde uživatel může napsat svoji odpověď, druhý Label slouží k oznámení, jestli odpověď byla správná, nebo ne. A na konec zde najdeme tlačítko, které vypisuje další slovo.



Obrázek 6 - Druhá scéna

Poslední třetí scéna obsahuje Label s počtem získaných bodů a tlačítko s funkcí program opakovat.



Obrázek 7 - Třetí scéna

Zde můžeme vidět využití prvku `Background`, který zapouzdřuje celou sadu dat potřebnou k vykreslení pozadí oblasti a je spíše stylizační, spolu se všemi použitými prvky a komponenty:

```
//Vytvoření všech používaných tlačítek
Button zkontrolovat = new Button( s: "zkontrolovat");
Button dalsi = new Button( s: "další");
Button zacinameButton = new Button( s: "Start");
Button zkusitz = new Button( s: "zkusit znovu");

//Tento štítek s textem obsahuje náhodný index z ArrayListu pro česká slova
Label cesky = new Label(cestina.get(index));

//Ostatní štítky s textem
Label pocetboduLabel = new Label();
Label kontrola = new Label();

//Textové pole pro naši odpověď
TextField textField = new TextField();

//Slouží k zobrazení všech určených komponentů v dané scéně
VBox vbox = new VBox();
StackPane layout = new StackPane();
VBox layout1 = new VBox();

//Vytvořené scény
Scene scene1 = new Scene(layout, w: 600, h: 400);
Scene scene2 = new Scene(vbox, w: 600, h: 400);
Scene scene3 = new Scene(layout1, w: 600, h: 400);

//Vytvoření pozadí s zelenou barvou
BackgroundFill background_fill = new BackgroundFill(Color.GREEN, CornerRadii.EMPTY, Insets.EMPTY);
Background background = new Background(background_fill);
```

Obrázek 8 - Použité prvky a komponenty

Mezi stylistické úpravy patří zvětšení tlačítek a jejich textu, popřípadě změnění barvy a typu textu. Pozadí je změněné na zelenou barvu.

```
//Nastavení velikosti tlačítka a textu v něm
zacinameButton.setMaxSize( w: 150, h: 90);
zacinameButton.setStyle("-fx-font-size: 2em; ");

//Nastavení typu textu a velikosti u určitého štítku s textem
cesky.setFont(new Font( s: "Arial", w: 24));
pocetboduLabel.setMaxSize( w: 300, h: 220);
pocetboduLabel.setFont(Font.font( s: "Verdana", FontWeight.BOLD, w: 25));

//Nastavení typu textu a velikosti u textového pole
textField.setFont(Font.font( s: "Verdana", FontWeight.BOLD, w: 20));
textField.setMaxSize( w: 250, h: 100);
```

Obrázek 9 - Stylistické úpravy

## 3.2. Implementace funkcí

Začneme od již už zmíněného tlačítka “Začínáme”, které má jednoduchou funkci, a to přepínat na jinou scénu. To v mém kódu vypadá takto:

```
//přidána funkce, kdy po stisknutí tlačítka "zaciname" se změní scéna na scénu dva
zacinameButton.setOnAction(e -> stage.setScene(scene2));
```

Obrázek 10 - tlačítko “Začínáme”

Další je tlačítko “Zkontrolovat”, které má dvě funkce. Pokud uživatel přeložil dané slovo správně, tlačítko nastaví text “správně” v Labelu pod ním a přičte prvek do ArrayListu pro počet bodů, pokud uživatel odpověděl nesprávně, text se v Labelu nastaví na “špatně” a dané slovíčko přidá do následující pozice v ArrayListu a to samé udělá s překladem. V obou se přičte prvek do ArrayListu pro počet opakování a tlačítko “Zkontrolovat” po zmáčknutí už nejde zmáchnout znovu.

```
zkontrolovat.setOnAction(e -> {
//Neznámá se vstupem z klávesnice
String odpoved = textField.getText();
//Pokud se vstup z klávesnice rovná německému překladu českého slova
if (odpoved.equals(nemcina.get(index))) {
//Nastaví se daný text a jeho barva
kontrola.setText("spravne");
kontrola.setTextFill(Color.LIME);
//Do ArrayListu "pocetBodu" a "pocetOpakovaniReset" se přičte element
pocetBodu.add(69);
pocetOpakovaniReset.add(69);
//Nastavení štítku s textem pro zobrazení počtu bodů
pocetboduLabel.setText("počet bodů: " + pocetBodu.size() + " z " + pocetOpakovani);
//Tlačítko "dalsi" bude zmáčknutelné
dalsi.setDisable(false);
//Tlačítko "zkontrolovat" bude nezmáčknutelné
zkontrolovat.setDisable(true);
}
```

Obrázek 11 – Funkce, pokud se rovná

```
//Pokud se vstup z klávesnice nerovná německému překladu
} else {
//Do ArrayListu "pocetOpakovaniReset" se přičte element
pocetOpakovaniReset.add(69);
//Nastaví se daný text a jeho barva
kontrola.setTextFill(Color.RED);
kontrola.setText("spatne");
//Do ArrayListu s českými i německými slovy se přičte element zodpovězený špatně a tím se jeho šance, že se objeví, zvýší
cestina.add(cestina.get(index));
nemcina.add(nemcina.get(index));
pocetboduLabel.setText("počet bodů: " + pocetBodu.size() + " z " + pocetOpakovani);
//Tlačítko "dalsi" bude zmáčknutelné
dalsi.setDisable(false);
//Tlačítko "zkontrolovat" bude nezmáčknutelné
zkontrolovat.setDisable(true);
}
});
```

Obrázek 12 – Funkce, pokud se nerovná

Tlačítko “dalsi” má také dvě funkce. Jedna nastane, pokud počet prvků v ArrayListu pro počet opakování se nerovná určitému počtu, v takovém případě se pouze vymaže text v

TextField a Labelu a Tlačítko “Zkontrolovat” bude možné opět zmáčknout a tlačítko “dalsi” naopak. V druhém případě, tedy když počet prvku bude roven, se přepne na třetí scénu.

```
dalsi.setOnAction(e -> {
    index = a.nextInt(cestina.size());
    //Po zmáčknutí nastaví se náhodný element do štítku s textem
    cesky.setText(cestina.get(index));
    //Pokud je pocet elementů v "pocetOpakovaniReset" roven "pocetOpakovani", po stisknutí tlačítka "dalsi" scéna se přepne na scénu 3
    if(pocetOpakovaniReset.size() == pocetOpakovani){
        stage.setScene(scene3);
    }else{
    //Pokud ne, text z textového pole a štítku se vymaže
        cesky.setText(cestina.get(index));
        kontrola.setText("");
        textField.setText("");
    //Tlačítko "zkontrolovat" bude zmáčknutelné
        zkontrolovat.setDisable(false);
    //Tlačítko "dalsi" bude nezmáčknutelné
        dalsi.setDisable(true);
    }
});
```

Obrázek 13 – Funkce tlačítka “dalsi”

Poslední je tlačítko “zkusitz”, které má program zpustit znovu. Při takovém zákroku je nutné vymazat prvky z ArrayListu pro počet bodů, počet opakování, česká i německá slovíčka. U ArrayListu pro slovíčka je samozřejmě nutné přidat původní prvky, tento krok je z důvodu, aby se na další pokus resetovala zvýšená šance na výskyt špatně zodpovězených slov. Dále se už jen vymaže text z TextField a Label a nastaví se které tlačítko jde či nejde zmáčknout.

```
zkusitz.setOnAction(c ->{
    //Po zmáčknutí se scéna změní spátky na scénu 2
    stage.setScene(scene2);
    //Vymaže všechny elementy z listu pro česká i německá slovíčka a přidá původní počet elementů
    cestina.clear();

    cestina.add("snídaně");
    cestina.add("snídat");
    cestina.add("zelenina");

    nemcina.clear();

    nemcina.add("Frühstück");
    nemcina.add("frühstücken");
    nemcina.add("das Gemüse");
    nemcina.add("das Plakat");

    //Text z textového pole a štítku se vymaže
    textField.setText("");
    kontrola.setText("");
    //tlačítko "zkontrolovat" bude zmáčknutelné a "dalsi" nezmáčknutelné
    zkontrolovat.setDisable(false);
    dalsi.setDisable(true);
    //Všechny elementy se z ArrayListů vymažou
    pocetOpakovaniReset.clear();
    pocetBodu.clear();
});
```

Obrázek 14 až 16 – Funkce tlačítka “zkusitz”

## Závěr

Celkový pohled z mé strany na finální verzi ročníkového projektu je takový, že se práce vyvedla. Zadání jsem úspěšně splnil a program bezchybně funguje. Program jde určitě dále rozvíjet a obohacovat o další funkce, například vytvořit pohodlný systém, kde sám uživatel zadá své slovíčka k procvičování. A mnoho dalšího.

Při práci jsem projevil opravdový zápal do dané problematiky a získal jsem mnoho užitečných zkušeností a vědomostí, které v budoucích projektech využiji.

## **Zdroje**

- [1] Oracle. (n.d.). *What is Java technology and why do I need it?* Java.com. Retrieved May 1, 2022, from [https://www.java.com/en/download/help/whatis\\_java.html](https://www.java.com/en/download/help/whatis_java.html)
- [2] JavaFX. (n.d.). Retrieved May 1, 2022, from <https://openjfx.io/>
- [3] 2000–2022 JetBrains s.r.o. (n.d.). *IntelliJ IDEA overview: IntelliJ idea*. IntelliJ IDEA Help. Retrieved May 1, 2022, from <https://www.jetbrains.com/help/idea/discover-intellij-idea.html>
- [4] Oracle. (2022, March 10). *Class ArrayList*. ArrayList (Java Platform SE 8 ). Retrieved May 1, 2022, from <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>
- [5] Oracle. (2020, June 24). *Class Scanner*. Scanner (java platform SE 7 ). Retrieved May 1, 2022, from <https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>
- [6] Oracle. (2022, March 10). *Class Random*. Random (java platform SE 8 ). Retrieved May 1, 2022, from <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>
- [7] Oracle. (n.d.). *The for statement*. The for Statement (The Java™ Tutorials > Learning the Java Language > Language Basics). Retrieved May 1, 2022, from <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>
- [8] Jenkov, J. (n.d.). *Java Tutorial*. Jenkov.com. Retrieved May 1, 2022, from <https://jenkov.com/tutorials/javafx/webview.html>

<https://www.javaprogrammingforums.com/computer-support/43945-i-am-trying-build-vocabulary-learning-program-need-urgently-help-solve-issues-method-etc-make-work.html>

<https://www.delftstack.com/howto/java/java-arraylist-of-ints/>

<https://www.geeksforgeeks.org/getting-random-elements-from-arraylist-in-java/>

JavaFX Java GUI Tutorial - 4 - Switching Scenes

<https://stackoverflow.com/questions/71887428/how-to-implement-java-methods-in-to-javafx>

[https://stackoverflow.com/questions/32624748/javafx-change-font-and size-in-a-textfield](https://stackoverflow.com/questions/32624748/javafx-change-font-and-size-in-a-textfield)

<https://stackoverflow.com/questions/61052676/how-to-change-color-of-text-in-javafx-label>

<https://www.geeksforgeeks.org/javafx-background-class/>

[https://upload.wikimedia.org/wikipedia/en/thumb/3/30/Java\\_programming\\_language\\_logo.svg/1200px-Java\\_programming\\_language\\_logo.svg.png](https://upload.wikimedia.org/wikipedia/en/thumb/3/30/Java_programming_language_logo.svg/1200px-Java_programming_language_logo.svg.png)

## Seznam obrázků

|   |    |
|---|----|
| Obrázek 1 – logo programovacího jazyka Java.....  | 7  |
| Obrázek 2 až 3 - Vytvoření ArrayListu.....        | 8  |
| Obrázek 4 – Funkce programu.....                  | 9  |
| Obrázek 5 - První scéna.....                      | 10 |
| Obrázek 6 - Druhá scéna.....                      | 10 |
| Obrázek 7 - Třetí scéna.....                      | 11 |
| Obrázek 8 - Použité prvky a komponenty.....       | 11 |
| Obrázek 9 - Stylistické úpravy.....               | 11 |
| Obrázek 10 - tlačítko “Začínáme”.....             | 12 |
| Obrázek 11 – Funkce, pokud se rovná.....          | 12 |
| Obrázek 12 – Funkce, pokud se nerovná.....        | 12 |
| Obrázek 13 – Funkce tlačítka ”dalsi”.....         | 13 |
| Obrázek 14 až 16 – Funkce tlačítka “zkusitz”..... | 13 |