

# Analýza, teoretický návrh a zpracování

Kvíz - Mafia: The City of Lost Heaven

Vendula Šmídová

Gymnázium Brno Vídeňská 47, příspěvková organizace

5.F

2021

Vedoucí práce: Ing. David Martínek



## Poděkování

Ráda bych touto cestou poděkovala vedoucímu mé bakalářské práce p. Ing. Davidu Martínkovi za jeho odborné vedení a cenné rady a připomínání časového rámce. Dále bych ráda poděkovala p. Mgr. Petru Šatnému, za krásná a přehledná videa o tvorbě GUI, je pro mě vždy inspirací. Také bych ráda poděkovala svému blízkému příteli Filipu Dobrovolnému za profesionální konzultace a rady, které mi po dobu mého řešení poskytoval.

# Abstrakt

Mým cílem, bylo vytvořit projekt, který provede triviální kvíz o hře Mafia: The City of Lost Heaven.

V této práci se budu zabývat analýzou a následnou praktickou implementací tohoto úkolu. Také se zde budu zabývat problémy, které při řešení vyvstaly a jak jsem je následně řešila, nebo řešit budu. V závěru bych se chtěla zabývat i otázkami, kam se projekt dále může ubírat.

## Klíčová slova

program, java, kvíz, Mafia, otázky, odpovědi, programování

# Abstract

My goal was to create a project that would play a trivial quiz about Mafia: The City of Lost Heaven.

In this thesis I will deal with the analysis and talk about the practical implementation of this task. I will also deal with the problems that arose during the solution and how I subsequently solved them, or how I will solve them later. In conclusion, I would like to address where the project can go next.

## Keywords

program, java, quiz, Mafia, questions, answers, programming

<b>Poděkování</b>	<b>2</b>
<b>Abstrakt</b>	<b>3</b>
<b>Klíčová slova</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>Keywords</b>	<b>4</b>
<b>Úvod</b>	<b>7</b>
<b>Analýza</b>	<b>8</b>
Vstupní soubor	8
Analýza struktury programu	8
Analýza grafického rozhraní	9
<b>Popis rozšíření oproti zadání</b>	<b>10</b>
Možnost odpovědí za pomocí textové pole	10
Klasifikace uživatelského skóre do kategorií	10
Hudba	10
Editor kvízu	11
<b>Praktická implementace</b>	<b>12</b>
Třída Kviz	12
Proměnná instance Kviz - vicHranic	12
Proměnná instance Kviz - otazky	12
Proměnná instance Kviz - otazka_data	12
Proměnná instance Kviz - index	12
Proměnná instance Kviz - bodyInt	12
Proměnná instance Kviz - celkemBodu	12
Konstruktor třídy Kviz	13
Metoda třídy Kviz - zobraz	13
Metoda třídy Kviz - spravnaVybrana	13
Metoda třídy Kviz - nacti	13
Metoda třídy Kviz - OkActionPerformed	15
Metoda třídy Kviz - main	16
Třída KvizovaOtazka	16
Proměnná instance KvizovaOtazka - otazka	16
Proměnná instance KvizovaOtazka - odpovedi	16

Proměnná instance KvizovaOtazka - body	16
Proměnná instance KvizovaOtazka - jeTextova	16
Metoda třídy KvizovaOtazka - vypis	16
Metoda třídy KvizovaOtazka - odpovediArray	16
Třída KvizovaOdpoved	16
Proměnná instance KvizovaOdpoved - odpoved	17
Proměnná instance KvizovaOdpoved - jeSpravne	17
Metoda třídy KvizovaOdpoved - vypis	17
Třída Hudba	17
Proměnná instance Hudba - clip	17
Konstruktor třídy Hudba	17
Metoda třídy pustHudbu	17
Třídy BodovaHranice	17
Proměnná instance BodovaHranice - hranice	18
Proměnná instance BodovaHranice - hodnoceni	18
Metoda třídy BodovaHranice - vypis	18
<b>Závěr</b>	<b>18</b>

# Úvod

Tato práce se bude zabývat tvorbou kvízu. Uživatel spustí kvíz s předem vygenerovanými otázkami ve kvízovém souboru, ty se mu potom budou náhodně ukazovat. Uživatel má vždy na výběr ze tří odpovědí, jedna z nich je správná, ostatní špatné. Při správné odpovědi se přičte určitý počet bodů, které jsou pevně dané a při špatné se nepřipíše nic. Uživatel po celou dobu běžení programu vidí svůj počet bodů v levém dolním rohu. Na konci kvízu se uživateli ukáže kolik bodů dosáhl z plného počtu. Program následně zhodnotí jak si uživatel vedl, podle toho, kolik měl bodů.

# Analýza

V této kapitole se budu zabývat elementárním návrhem této aplikace. V první řadě zde popíšu vstupní formát kvízu a osvětlím jeho význam a jeho budoucí rozvoj.

## Vstupní soubor

```
Název kvízu
??Otázka??
#počet bodů
+správná odpověď
-špatná odpověď
-špatná odpověď
?= Otázka, která očekává odpověď textovým polem
#Počet bodů
+Správná odpověď
!33%No moc ses nepředvedl!
!66%Celkem se orientuješ, ale mohlo by to být
lepší!
!100%Jde vidět, že jsi na Mafii vyrůstal. Pan
Salieri by tě určitě přivítal do rodiny!
```

První řádek popisuje Název Kvízu. Za dvěma otazníky program očekává kvízovou otázku s volbou z vypsanych možností. Za symbolem “#” je číslem napsaný počet bodů za konkrétní otázku. Za symbolem “+” je správná odpověď, za symbolem “-” špatná odpověď. Za množinou symbolů “?” program očekává otázku, kde se odpověď zadává textovým polem. Za symbolem “!” je vždy napsaná bodová hranice a jak program zhodnotí uživatelské vstupy.

## Analýza struktury programu

V programu se nachází základní třída Kviz. Tato třída je základním kamenem celého programu. Obsahuje metodu main, která se využívá pro spuštění programu. Třída dědí z grafické knihovny a díky tomu je tato třída základním oknem celého programu. Stará se také o načtení kvízu ze souboru podle výše uvedených pravidel. V případě že dojde k



chybě v souboru, program vyhodí chybové hlášení a program po zavření okna skončí. Třída Kviz obsahuje číselný ukazatel dosaženého počtu bodů. Dále tato třída obsahuje kolekci otázek, ta je reprezentována třídou KvizovaOtazka. KvizovaOtazka obsahuje kolekci odpovědí, druh otázky a počet bodů, které uživatel obdrží při jejím správném zodpovězení. Odpověď v otázce je reprezentována třídou KvizovaOdpoved, která obsahuje text odpovědi a informaci, zda-li je správná. Následně třídy KvizovaOtazka a KvizovaOdpoved obsahují funkce vypis, které slouží pro vypsání této reprezentace zpět do textové podoby. Také jsem nově přidala třídu Hudba, ve které je krátký kód, který při spuštění kvízu spustí soubor s hudbou, který následně bude po celou dobu kvízu hrát, dokud jej uživatel neukončí.

## Analýza grafického rozhraní

Tento program využívá dvou grafických prostředků. Prvním je samotné okno kvízu. Toto okno obsahuje jméno otázky. V názvu okna se nachází název kvízu. Pod jménem otázky je textové pole, které je aktivní v případě, že aktuální otázka je takového druhu, že očekává odpověď textovým polem. V případě, že se nejedná o tento typ otázek, je toto pole neaktivní a uživatel s ním nemůže pracovat. Pod textovým polem se nachází seznam, ze kterého lze vybrat odpověď. V dolní části se nachází tlačítko "Ok", pomocí kterého označí uživatel zvolenou nebo vypsanou odpověď za svůj finální výběr. V okamžiku, kdy odpověď je správná, vyskočí dialogové okno, které uživateli sdělí správnost jeho odpovědi a jeho aktuální počet bodů. V případě, že jeho odpověď správná nebyla, vyskočí dialogové okno, které jej na tuto skutečnost upozorní. Ve chvíli kdy uživatel dojde na konec kvízu, vyskočí finální dialog s celkovým vyhodnocením kvízu. V okamžiku, kdy ve zdrojovém souboru jsou definovány stupnice úspěšnosti, zařadí kvíz uživatele do jeho příslušné kategorie a tuto kategorii zobrazí.

## Popis rozšíření oproti zadání

Svůj program jsem se rozhodla rozšířit o následující:

### Možnost odpovědí za pomoci textové pole

Toto rozšíření umožňuje uživateli zadávat kromě pouhých prachsprostých otázek typu “a/b/c/.../z”, je možné také inovativně zadávat odpovědi vlastními slovy. Jsem s to, že toto rozšíření, je ač poměrně jednoduché, velice efektivní. Uživatel má potom pocit jakési volnosti a dynamičnosti. Co ovšem neví je, že pokud napíše jen jedno písmenko či číslici špatně, prohraje vše.

Realizace tohoto rozšíření je provedena pomocí nového identifikátoru řádku v kvízovém souboru “?=” a upravením kódu, například pomocí atributu “jeTextova”, a upravením uživatelského rozhraní přidáním textového pole pod název otázky.

### Klasifikace uživatelského skóre do kategorií

Rozšíření slouží jakožto jakási forma zpětné vazby pro uživatele. Uživatel se dozví, jak je na tom. A můžu se cítit dobře nebo špatně nad svým výsledkem na své cestě životem. Máme n kategorií, podle toho, jaká byla naše procentuální úspěšnost v tomto kvízu.

Formát vstupního souboru byl rozšířen o identifikátor “!”. Pomocí něhož lze definovat horní bodovou hranici kategorie a její název. Taktéž byl kód upraven, aby na začátku svého běhu spočítal celkový součet bodů všech otázek, aby bylo možné na konci běhu programu, spočítat procentuální úspěšnost a na základě ní, zařadit uživatele do příslušné kategorie.

### Hudba

Rozšíření, které po spuštění kvízu začne hrát hudbu, která vás po celé době řešení kvízu bude provázet.

## Editor kvízu

Rozšíření, které umožňuje vytvořit si vlastní kvíz a pobavit tak rodinu i přátele. Uživatel bude mít kontrolu nad tím, jaké otázky a odpovědi zadává. Bude si připadat velice chytrý v porovnání s ostatními.

Pro zhmotnění tohoto rozšíření dojde k vytvoření separátní aplikace, kde pomocí série dialogových oken, bude možné upravovat a přidávat otázky a odpovědi do existujících nebo i nových kvízů za pomoci GUI.

# Praktická implementace

## Třída Kviz

Tato třída je zodpovědná za základní běh programu. Načítá kvíz a zobrazuje uživatelské rozhraní.

### Proměnná instance Kviz - vicHranic

Tato kolekce obsahuje jednotlivé bodové hranice v podobě BodoveHranice. K inicializaci této kolekce dochází v konstruktoru třídy Kviz. K naplnění této kolekce dochází v metodě nacti.

### Proměnná instance Kviz - otazky

Tato kolekce obsahuje jednotlivé kvízové otázky v podobě KvizovaOtazka. K inicializaci této kolekce dochází v konstruktoru třídy Kviz. K naplnění této kolekce dochází v metodě nacti.

### Proměnná instance Kviz - otazka\_data

Tato proměnná obsahuje instanci aktuálně zobrazované otázky. K nastavení této proměnné dochází v metodě zobraz.

### Proměnná instance Kviz - index

Tato proměnná ukazuje na aktuální polohu uživatele v kvízu. K její změně dochází v metodě OkActionPerformed, která se spustí v okamžiku, kdy uživatel klikne na tlačítko ok.

### Proměnná instance Kviz - bodyInt

Tato proměnná ukládá uživatelův dosažený počet bodů.

### Proměnná instance Kviz - celkemBodu

Tato proměnná obsahuje celkový počet bodů. K naplnění této proměnné dochází v metodě nacti.

## Konstruktor třídy Kviz

Tento konstruktor nastaví pole otázek a pole bodových hranic.

## Metoda třídy Kviz - zobraz

Tato metoda načte aktuální otázku do proměnné `otazka_data`. Následně zobrazí jméno otázky, vyčistí textové pole a v závislosti na druhu otázky se aktivuje správné pole pro odpověď.

## Metoda třídy Kviz - spravnaVybrana

Tato metoda kontroluje, jestli je moje vypsaná odpověď správná. Pokud otázka vyžaduje odpověď textovým polem, program porovná uživatelskou odpověď se správnou odpovědí, nehledě na malá či velká písmena. Pokud očekává jako odpověď výběr z nabízených odpovědí, zjistí, která odpověď je vybraná. Pokud není žádná, vyskočí dialog, který uživatele na tuto skutečnost upozorní. Pokud je vybrána odpověď, vrátí, jestli je správná.

## Metoda třídy Kviz - nacti

Tato metoda při spuštění vyhodí dialog, který informuje uživatele, že se načítá soubor. Poté se pokusí otevřít soubor. Pokud se mu to nepodaří, vyhodí dialogové okno, které uživatele s touto skutečností obeznámí. Pokud je pokus úspěšný, pokusí se přečíst první řádek. Pokud se to podaří, kvízu se nastaví jméno stejné, jako tento řádek. Pokud se to nepovede, vyskočí dialog, který uživateli sdělí, že vstupní soubor neobsahuje název kvízu.

Následně program čte soubor řádek po řádku a mohou nastat následující varianty:

1. Řádek začíná “??” - při této variantě program vytvoří novou otázku, pokud již má rozpracovanou jinou otázku, zkontroluje, zda-li otázka má odpovědi. A zkontroluje, pokud je otázka textová, zda má pouze jednu odpověď. Pokud vše proběhlo v pořádku, zpřehází pořadí možných odpovědí. Poté si otázku “odloží”. Program vytvoří novou prázdnou otázku, otázka je typu výběr, s názvem uvedeným v souboru (bez prvních dvou znaků, ty slouží jako identifikátory typu záznamu v souboru).
2. Řádek začíná “?=” - při této variantě program vytvoří novou otázku, pokud již má rozpracovanou jinou otázku, zkontroluje, zda-li otázka má odpovědi. A zkontroluje, pokud je otázka textová, zda má pouze jednu odpověď. Pokud vše proběhlo v pořádku, zpřehází pořadí možných odpovědí. Poté si otázku “odloží”. Program vytvoří novou prázdnou otázku, otázka je typu textové pole, s názvem uvedeným v souboru (bez prvních dvou znaků, ty slouží jako identifikátory typu záznamu v souboru).
3. Řádek začíná “#” - při této variantě program nejprve testuje, zda započala nějaká otázka. Pokud ne, vypíše chybové hlášení do dialogového okna. Pokud ano, pokusí se načíst zadaný počet bodů (bez prvního znaku, ty slouží jako identifikátory typu záznamu v souboru). Pokud za symbolem “#” není číslo, vypíše dialog s chybovým hlášením.
4. Řádek začíná “+” - zkontroluje, jestli je momentálně nějaká otázka rozpracovaná. Pokud ne, vyhodí dialog s chybovým hlášením. Pokud ano, vytvoří novou správnou odpověď, která se pojmenuje zbytkem řádku (bez prvního znaku, ty slouží jako identifikátory typu záznamu v souboru).
5. Řádek začíná “-” - zkontroluje, jestli je momentálně nějaká otázka rozpracovaná. Pokud ne, vyhodí dialog s chybovým hlášením. Pokud ano, vytvoří novou špatnou odpověď, která se pojmenuje zbytkem řádku (bez prvního znaku, ty slouží jako identifikátory typu záznamu v souboru).

6. Řádek začíná “!” - zkontroluje, jestli je v řádku přítomen symbol “%”. Pokud ne, vyhodí dialog s chybovým hlášením. Pokud ano, text mezi vykřičníkem a procentem se použije po převedení na číslo, jako proměnná hranice v nové třídě BodovaHranice. Zbytek řádku za symbolem procento, se využije jako proměnná hodnocení ve třídě BodovaHranice.

Po projití celého souboru zkontroluje, jestli soubor obsahoval nějaké otázky. Pokud ne, vyhodí dialog s chybovým hlášením, pokud ano, pokračuje dále. Zkontroluje, zda-li rozpracovaná otázka má odpovědi. A zkontroluje, pokud je otázka textová, zda má pouze jednu odpověď. Pokud vše proběhlo v pořádku, zpřehází pořadí možných odpovědí. Poté si otázku “odloží”. Následně zpřehází pořadí otázek. Nakonec seřadí kolekci vicHranic dle proměnné hranice v jednotlivých instancích.

#### Metoda třídy Kviz - OkActionPerformed

V okamžiku stlačení tlačítka “Ok” se spustí. Nejprve zkontroluje, pokud se jedná o otázku typu výběr, zda-li byla vybrána odpověď. Pokud ne, zobrazí dialog s chybovým hlášením. Poté zkontroluje, pokud se jedná o otázky, která očekává jako odpověď textové pole, zda-li byla nějaká odpověď zadána. Pokud ne, vyhodí dialog s chybovým hlášením.

Pokud uživatel vybral správnou odpověď, vyhodí dialog, který to uživateli sdělí. Poté se přičte určitý počet bodů náležící konkrétní otázce, a toto číslo se také zobrazí v levém dolním rohu.

Poté kontroluje, jestli v kolekci jsou ještě další otázky. Pokud ano, přejde na další otázku a pomocí funkce zobraz přepíše display. Pokud ne, jsme tedy na konci kvízu. Program uživateli pográtuluje (pomocí dialogu) a vypíše jeho počet bodů a procentuální úspěšnost toho, jak moc se mu kvíz povedl. Následně, dle úspěšnosti, pokud nebyly načteny žádné BodovaHranice, podle natvrdo napsaných kritérií uživateli zobrazí slovní hodnocení jeho výkonu. V případě, že došlo k nějakému načtení BodovaHranice, vybere a zobrazí slovní hodnocení z načtených BodovaHranice.

Metoda třídy Kviz - main

Program se spouští touto metodou. Nejprve dojde k načtení a spuštění hudby za pomoci třídy Hudba. Následně dochází k načtení kvízových dat za pomoci metody nacti. V případě, že tato metoda úspěšně vykonala svou funkci, za pomoci metody zobraz nastaví grafickým prvkům odpovídající hodnoty. Za předpokladu, že dojde k chybě během načítání dat, končí program.

## Třída KvizovaOtazka

Tato třída, jak již název napovídá, reprezentuje otázku v kvízu.

Proměnná instance KvizovaOtazka - otazka  
Název otázky.

Proměnná instance KvizovaOtazka - odpovedi  
Kolekce odpovědí.

Proměnná instance KvizovaOtazka - body  
Kolik bodů je za danou otázku.

Proměnná instance KvizovaOtazka - jeTextova  
Pravdivostní hodnota rozhodující o druhu otázky.

Metoda třídy KvizovaOtazka - vypis

Vypíše na obrazovku podobu této kvízové otázky, tak, jak by byla napsaná ve vstupním souboru.

Metoda třídy KvizovaOtazka - odpovediArray

Přetransformuje odpovědi otázky na formát, který je použitelný pro výběrové pole GUI.

## Třída KvizovaOdpoved

Tato třída prezentuje odpověď ke kvízové otázce.



Proměnná instance KvizovaOdpoved - odpoved

Uchovává ve svém lůně odpovědi na veškeré otázky lidstva. (Nebo možná spíš jen na určitou kvízovou).

Proměnná instance KvizovaOdpoved - jeSpravne

Říká, jestli je správně.

Metoda třídy KvizovaOdpoved - vypis

Vypíše na obrazovku podobu této kvízové odpovědi, tak, jak by byla napsaná ve vstupním souboru.

Třída Hudba

Tato třída spustí soubor s hudbou.

Proměnná instance Hudba - clip

Uchovává instanci systémové přehrávače.

Konstruktor třídy Hudba

Pokusí se načíst soubor s hudbou a nastavit proměnnou clip. V případě, že se mu to nepovede, vyhodí uživateli varování, ale program bude pokračovat dál.

Metoda třídy pustHudbu

V případě, že se povedlo načíst hudbu, tak ji spustí.

Třídy BodovaHranice

Tento objekt uchovává informaci o bodové hranici.

Proměnná instance BodovaHranice - hranice

Uchovává procentuální bodovou hranici.

Proměnná instance BodovaHranice - hodnoceni

Uchovává výsledné hodnocení.

Metoda třídy BodovaHranice - vypis

Vypíše bodovou hranici, jak byla ve vstupním souboru.

## Závěr

Během realizace tohoto projektu jsem se setkala s řadou nesnází. Největší z nich bylo správně si naplánovat práci a rozdělit ji na dílčí úkoly. S tímto problémem se potýkám v životě studenta prestižního, výběrového, elitního gymnázia velice často, je to v podstatě můj denní chléb.

Co se samotného řešení týče, měla jsem pocit, jakoby mě hodiny dostatečně nepřipravily na řešení tohoto úkolu, a tak jsem byla nucena některé věci vyhledávat na internetu.

Svoji práci bych zhodnotila nejméně jako dostatečnou. Myslím si, že existují i horší řešení, a že to mé přeci jen funguje, i když jsem jej odevzdala s drobným zpožděním. Také bych ráda uvedla na pravou míru, jak se to má s mým editorem otázek. Bohužel, už mi na toto rozšíření nevyzbyl čas, ale v budoucnu budu zpět k tomu, abych toto rozšíření dokončila. Tato práce je pro mě další ukázkou toho, že se programování jako celku už po gymnáziu v životě nedotknu.