

# Semestrální práce

Algoritmizace a programování 2

Malčík Tomáš

Algoritmizace a programování 2

2020

## Obsah

**Obsah**

[Semestrální práce 1](#__RefHeading___Toc326_1732377548)

[Obsah 2](#__RefHeading___Toc328_1732377548)

[Zadání práce 3](#__RefHeading___Toc330_1732377548)

[Návrh řešení 4](#__RefHeading___Toc332_1732377548)

[Objektový model 5](#__RefHeading___Toc334_1732377548)

[Popis vstupních souborů a výstupu do databáze 6](#__RefHeading___Toc336_1732377548)

[Struktura „students.txt“ 6](#__RefHeading___Toc338_1732377548)

[Struktura „test1.txt“ 6](#__RefHeading___Toc340_1732377548)

[Testování aplikace 7](#__RefHeading___Toc342_1732377548)

[Špatně zadané údaje 7](#__RefHeading___Toc344_1732377548)

[Nevyplnění otázek 8](#__RefHeading___Toc346_1732377548)

[Nenalezení souboru s otázkami či se studenty 8](#__RefHeading___Toc348_1732377548)

[Nepřipojení do databáze 8](#__RefHeading___Toc350_1732377548)

[Funkční požadavky 9](#__RefHeading___Toc352_1732377548)

## Zadání práce

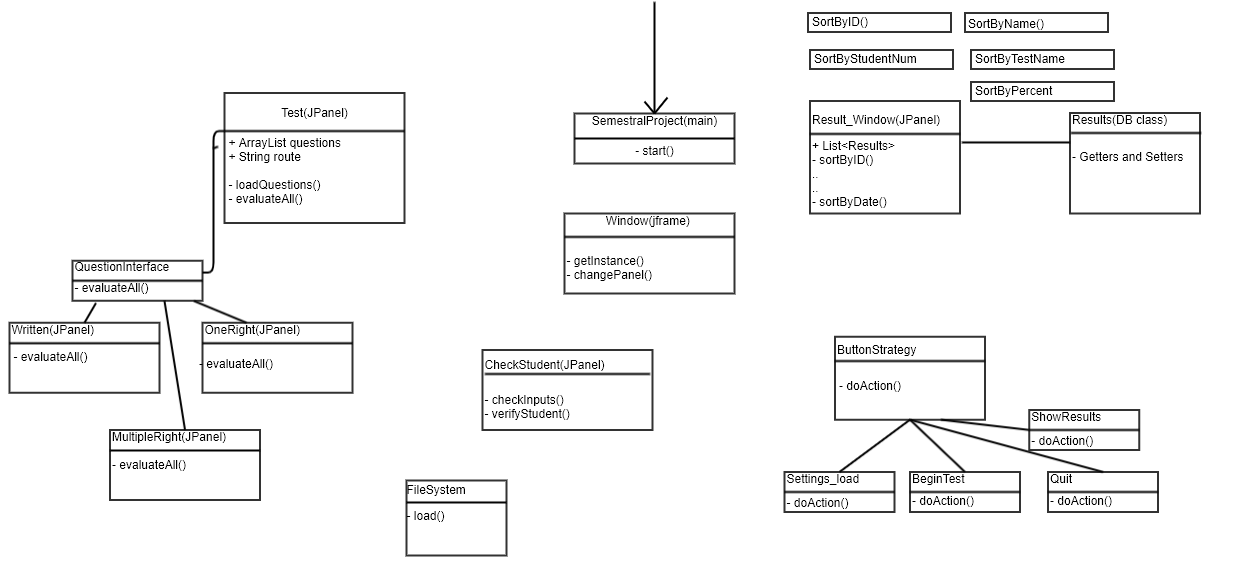
Vytvořte program pro školu, ve kterém si žáci budou moci vypracovávat testy. Každý žák se bude muset autorizovat zadáním svého jména a studentského čísla, ty se poté ověří s daty v souboru se studenty. Testy budou zadávány ve formě souboru, který bude obsahovat text otázky a jednotlivé odpovědi oddělené speciálním znakem. Program má obsahovat 3 typy otázek – písemné, s jednou správnou odpovědí a s více správnými odpovědmi. Dále si student bude moci prohlédnout výsledky testů, které budou mít možnost řazení podle všech údajů. Výsledky testů se budou ukládat do databáze libovolného typu.

Pro komunikaci s uživatelem použijte grafické rozhraní. Vstup od uživatele ošetřete tak, aby jeho studentské číslo náleželo tvaru „MXXXXXXXX“, kde X reprezentuje číslo od 0 do 9.

## Návrh řešení

1. Načíst otázky ze souboru
2. Ověření studenta
   1. Zadat jméno, příjmení a studentské číslo
      1. Ověřit tvar studentského čísla
      2. Ověřit tvar jména a příjmení
   2. Ověřit se studenty v souboru
3. Zpracování otázek
   1. otázka s jednou správnou odpovědí
   2. otázka s více možnými odpověďmi
   3. otázka se slovní odpovědí
4. Začít test
5. Vyhodnotit test
6. Uložit do databáze
7. Seřadit výsledky
   1. podle pořadí testu
   2. podle jména testu
   3. podle studentského čísla
   4. podle procent
   5. podle data
   6. podle času stráveného nad testem

## Objektový model



- Model obsahuje pouze ty hlavní třídy, které projekt obsahuje. Další, co se v projektu nachází jsou pouze nadstavby, či rozšíření.

## Popis vstupních souborů a výstupu do databáze

Projekt má pouze 2 vstupní soubory. K projektu je nutno mít soubor students.txt a test1.txt. Tyto soubory jsou přiloženy v ZIP souboru s projektem.

Lokace těchto souborů je „[C://Users/](../../../..//Users/)<uzivatel>/Semestral/test/..“.

### Struktura „students.txt“

<jméno a příjmení>#<studentské číslo>. Příklad: Tomáš Malčík#M19000059. Tento soubor slouží pro ověření, zda-li je student doopravdy studentem školy. Program se nadále stará o to, aby data byly ve správném formátu.

### Struktura „test1.txt“

<typ>#<text otázky>#<odpověď1>#<odpověď2>#<odpověď3>#<odpověď4>#<správně>

Příklad:

0#Test….#1#2#3#4#0. Tento soubor obsahuje otázky, uveden je text, odpovědi a typ. Typem se myslí jestli je písemná, má jednu správnou odpověď či více správných odpovědí.

## Testování aplikace

Jelikož aplikace nezískává od uživatele více dat než je jeho jméno a studentské číslo, bylo testování pojato takto:

### Špatně zadané údaje

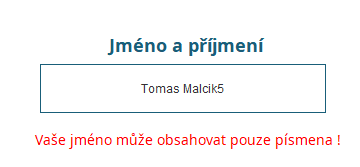
Ošetření údajů jsem provedl pomocí jLabelu. Pokud uživatel zadá jméno či studentské číslo ve špatném tvaru, vypíše se chybová hláška do jLabelu.

Správný tvar:

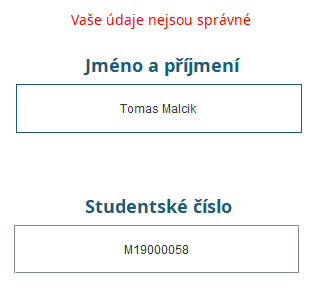
Jméno musí obsahovat pouze písmena.

Studentské číslo musí být ve tvaru (A-Z)XXXXXXXX, kde X reprezentuje číslo.

Testy:

  
Obrázek 1: Špatně zadané uživatelské jméno

  
Obrázek 2: Špatně zadané studentské číslo

  
Obrázek 3: Neexistující student

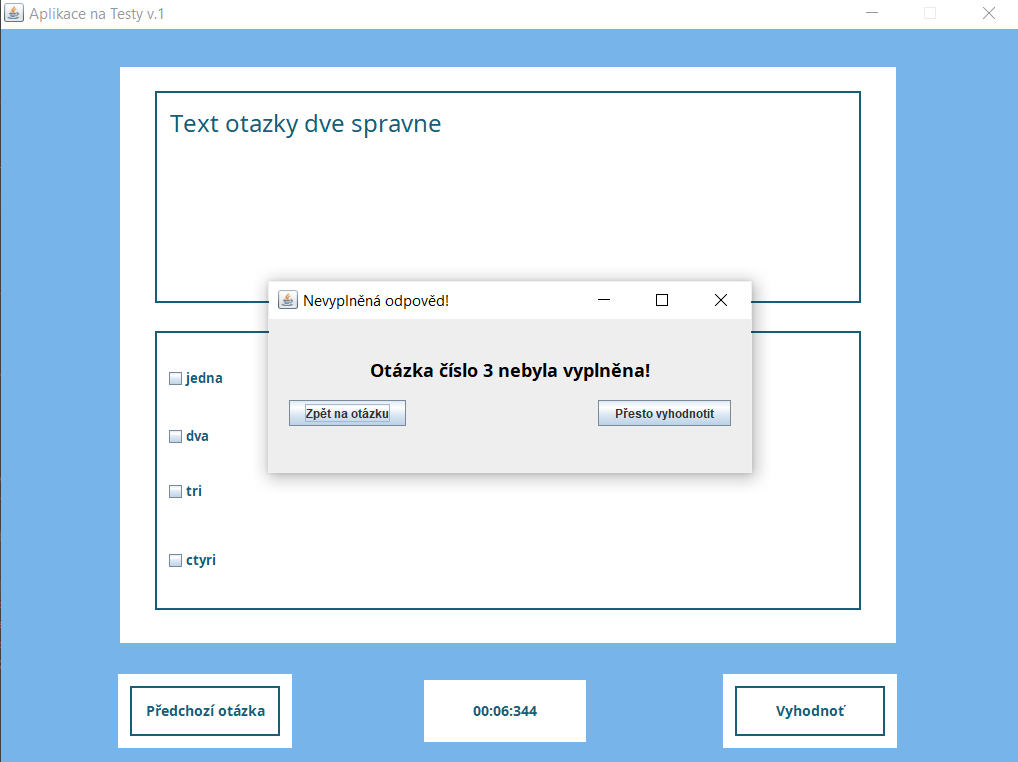
### Nevyplnění otázek

Pokud uživatel nevyplní otázku, tak se mu započítá při odevzdání jako chybná, při odevzdání se objeví okno, které zahlásí, že otázka X nebyla vyplněna

Test:

Nevyplnil jsem otázku 3:

Program dává možnost se k otázce vrátit, nebo vyhodnotit i přes to, že existuje nezodpovězená otázka

  
Obrázek 4: Nevyplněná otázka

### Nenalezení souboru s otázkami či se studenty

Pokud FileSystem nenalezne soubor, vyhodí výjimku, program avšak nespadne.



### Nepřipojení do databáze

Pokud se nepřipojíme k databázi, program nebude vůbec fungovat, jelikož by jeho některé funkce (ukládání do DB, načítání z DB) nefungovaly.

## Funkční požadavky

1. Javadoc - každá trieda a metóda musí mať popis (netreba pre evidentné konštruktory, gettre, settre a toString)
2. Menu, ktoré umožní opakovaný výber funkcií aplikácie a ukončenie aplikácie
3. Prehľadný (sformátovaný) výpis výsledkov na konzolu - použite String.format() a StringBuilder
4. Načítanie vstupných dat zo súboru
5. Zápis výstupných dat do súboru
6. Možnosť práce s textovými aj binárnymi súbormi
7. Tri balíčky:  a. ui – trieda Main s hlavným programom

b. app – triedy tvoriace logiku aplikácie - modely, kontrolery

c. utils – pomocné triedy napr. vlastné výnimky, rozhrania

1. Vlastni interface
2. Použitie java.time API pre prácu s časom
3. Použitie kontainerovej triedy jazyka Java (ArrayList, LinkedList, HashMap ...)
4. Aspoň dve možnosti triedenia s využitím rozhraní Comparable a Comparator
5. Použitie regulárneho výrazu
6. Ošetrenie vstupov, aby chybné vstupy nespôsobili ukončenie programu - pomocou existujúcich prípadne vlastných výnimiek
7. Vhodné ošetrenie povinne ošetrovaných výnimiek
8. Použitie Vami vybranej externej knižnice (audio, poslanie mailov, práca s obrázkom, junit testovanie ...)
9. Vítaná bude GUI

Tabulka obrázků

[Obrázek 1: Špatně zadané uživatelské jméno 7](#__RefHeading___Toc344_1732377548)

[Obrázek 2: Špatně zadané studentské číslo 7](#__RefHeading___Toc344_1732377548)

[Obrázek 3: Neexistující student 7](#__RefHeading___Toc344_1732377548)

[Obrázek 4: Nevyplněná otázka 8](#__RefHeading___Toc344_1732377548)