

Slovenská technická univerzita v bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

SPRÁVA O REALIZÁCIÍ PROJEKTU
Student Challenger

Adam Chmara
2017/2018
Objektovo orientované programovanie

Obsah

1. Zámer projektu	3
2. Štruktúra systému (diagram)	3
3. Kritéria hodnotenia	4
3.1 Naplnenie zámeru	4
3.2 Splnené programové kritéria	4
3.2.1 Dedenie a prekonávanie	4
3.2.2 Zapuzdrenie	4
3.2.3 Polymorfizmus	4
3.2.4 Agregácia	4
3.2.5 Návrhový vzor Observer	4
3.2.6 Vlastná výnimka	4
3.2.7 Oddelenie GUI od ap. logiky	4
3.2.8 Multithreading	5
3.2.9 Generickosť	5
3.2.10 RTTI	5
3.2.11 Lambda výrazy	5
3.2.12 Organizácia kódu	5
3.2.13 Dokumentácia	5
4. Zoznam odovzdaných pracovných verzií	5
5. Popis používateľského rozhrania	5
5.1 Testy	5
5.2 Profil	6
5.3 Ladderboard	6
5.4 Vytvor test	6

1. Zámer projektu

Zámer projektu je vytvoriť e-learning software, ktorý by umožnil študentom jednoduché a efektívne vzdelávanie sa, zároveň podnietil a motivoval študentov k opätovnému používaniu za pomoci jednoduchých princípov hry a súťaživosti.

Študent si po vytvorení profilu zvolí predmet kde môže plniť úlohy a testy, tzv. výzvy. Úlohy a testy sú rôzne náročné a špecifikované (čas, nápovedy, multiple-choice...). Podľa obtiažnosti a úspešnosti riešenia sa študentovi pripisujú body, ktoré ho umiestňujú do celkového rebríčka všetkých študentov. Za plnenie špeciálnych či mimoriadne náročných úloh (napr. tri testy nad 95% po sebe) študent získava trofeje, ktoré sa mu taktiež pripisujú do profilu a ovplyvňujú poradie v rebríčku. V profile študenta sú zobrazené údaje ako skill-level študenta (začiatočník, pokročilý...), celkové body, odznaky, počet splnených úloh a testov, percentuálna úspešnosť v jednotlivých predmetoch, a mnohé iné. Získané trofeje či iné významné udalosti všetkých užívateľov budú vypisované v hlavnom menu.

Výsledkom tohto projektu bude softvér ktorý posilní motiváciu k štúdiu za pomoci jednoduchých princípov hry a súťaživosti ako aj prehľad učiteľa o výsledkoch a výkone študentov, či jednoduchá implementácia nových testov a trofejí učiteľom.

2. Štruktúra systému (diagram)

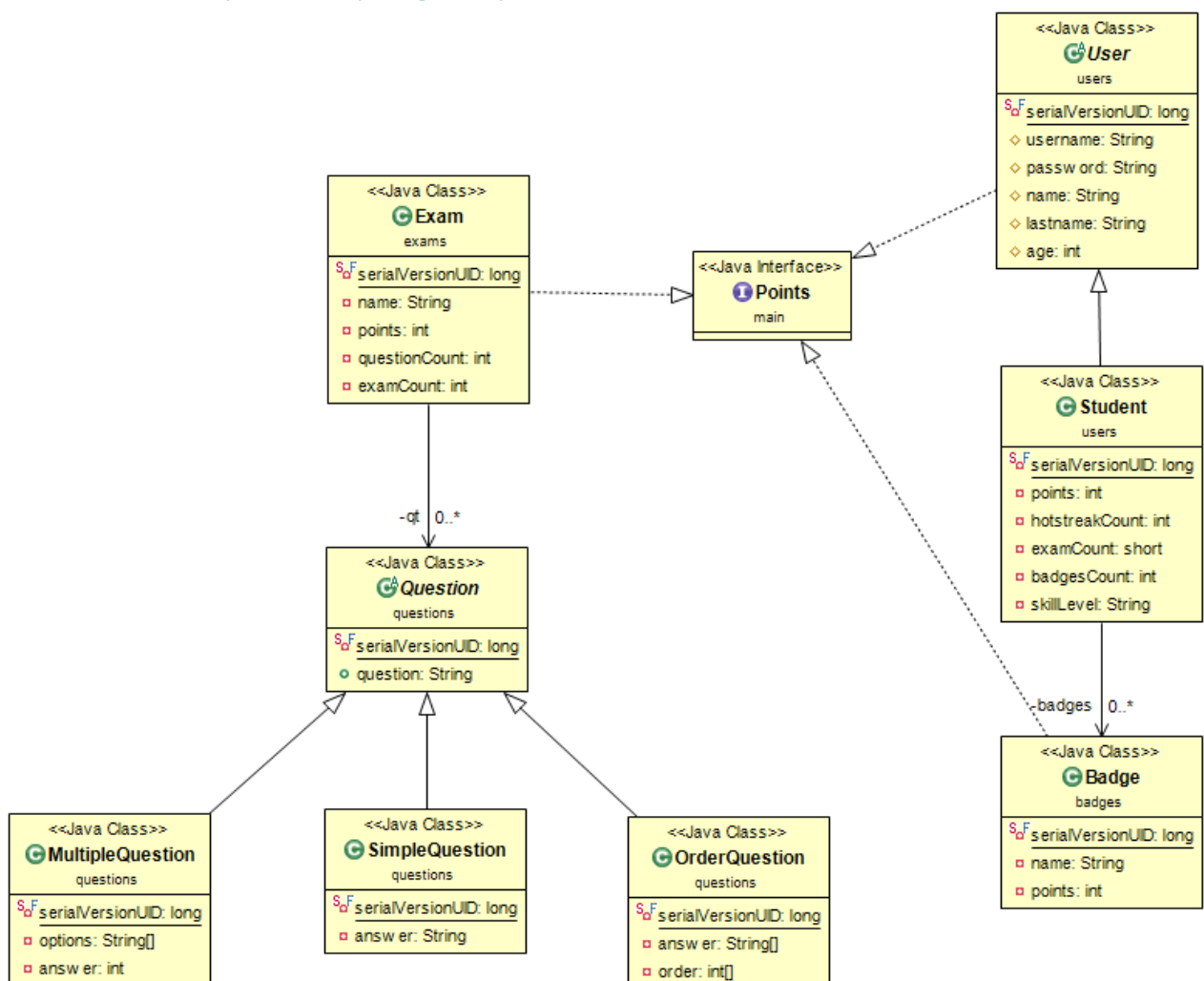


Diagram zobrazuje štruktúru a hierarchiu hlavných tried projektu. V štruktúre sú vyznačené vzťahy dedenia (Question -> SimpleQuestion...), agregácie (Exam -> Question, Student -> Badge), či jednotlivých logických spojení tried a rozhraní. V diagrame sú znázornené len logické časti programu bez prislúchajúcich metód pre prehľadnosť.

3. Kritéria hodnotenia

3.1 Naplnenie zámeru

Projekt splnil hlavné ciele a funkcionality stanovené v zámere, s výnimkou vynechania niektorých repetitívnych častí ako napr. užívateľské štatistiky či hlbšia kategorizácia testov. Počas realizácie projektu boli použité všetky požadované kritéria a niekoľko ďalších. Program je plne funkčný s možnosťou ďalšieho vývoja.

3.2 Splnené programové kritéria

3.2.1 Dedenie a prekonávanie

Program využíva dedenie v abstraktnej triede *Question* z ktorej dedia triedy *SimpleQuestion*, *MultipleQuestion*, *OrderQuestion*. Každá z týchto tried prekonáva metódy triedy *Question*. Ďalšie dedenie je v abstraktnej triede *User* a podtriede *Student*. Tak isto tu je možnosť rozšírenia dedenia, v prípade potreby, ako pri triede *Question* tak aj pri *User* či *Exam*. Prekonávanie je taktiež použité v *Threads* či ako súčasti iných metód.

3.2.2 Zapuzdrenie

Zapuzdrenie je využité vo všetkých triedach projektu podľa potreby. Údaje a premenné tried typu *private* a ich hodnoty dostávame pomocou *getters*, *setters* či iných metód.

3.2.3 Polymorfizmus

Polymorfizmus využívame pri *overloadingu* a *overridingu* metód v rôznych triedach. V triede *Exam* pri metóde *addQuestion* a *startExam*, tak isto aj v mnohých metódach triedy *Controller*.

3.2.4 Agregácia

V triede *Exam* implementujeme objekt triedy *Question*, tak isto aj v triede *Student* objekt triedy *Badge*.

3.2.5 Návrhový vzor Observer

V programe je implementovaný návrhový vzor *Observer* ktorý využívame pri skúmaní stavu a hodnoty premenných spojených s *Badges*.

3.2.6 Vlastná výnimka

V metóde *examNames* (Controller 321. riadok) ošetrujeme mimoriadny stav pomocou vlastnej výnimky ktorá vypíše *Error.message* v prípade prázdneho zoznamu testov.

3.2.7 Oddelenie GUI od ap. logiky

Používateľské rozhranie je v rámci možností oddelené od aplikačnej logiky pomocou triedy *Controller* ktorá spracováva zmeny a prenáša ich dynamicky na GUI, či pomocou volania statických prvkov rozhrania.

3.2.8 Multithreading

V GUI triedach *ExamGUI* a *CreateExam* sa nachádza Thread ktorý sa za pomoci *SwingWorker*a vykonáva paralelne v pozadí programu, tak isto aj hlási dokončenie procesu vykonávania. Ďalší thread implementujeme pri metóde **TimerTask** v triede *Controller*.

3.2.9 Generickosť

V triede *Controller* pri metódach serializácie/deserializácie ale aj iných využívame generickosť napríklad pri tvorení *List<User>* ktorý obsahuje objekty *Student*.

3.2.10 RTTI

RTTI využívame napríklad v triede *BadgeObserver* kde pomocou slova *instanceof* zisťujeme či je objekt potrebnej triedy a na základe toho vykonáme príslušnú metódu.

3.2.11 Lambda výrazy

Lambda výrazy využívame pri tvorbe ActionListenerov v GUI triedach obsahujúcich tlačidlá pre prehľadnosť a jednoduchosť kódu.

3.2.12 Organizácia kódu

Program je prehľadne zorganizovaný podľa funkčných a logických častí, kód jednotlivých tried je prehľadne zorganizovaný s vhodným pomenovaním metód a premenných. Triedy sú usporiadané do package-ov podľa funkcionality.

3.2.13 Dokumentácia

V prílohe programu je vygenerovaný *Javadoc* s popisom potrebných premenných a metód.

4. Zoznam odovzdaných pracovných verzií

Všetky verzie programu sú dostupné na GitHubu.

1. Pracovná verzia (10.4.2018)

Program pokrýval väčšinu zámeru.

2. Finálna verzia (24.4.2018)

Program pokrýva všetky stanovené funkcionality s priloženým UML diagramom a dokumentáciou.

5. Popis používateľského rozhrania

Celý program sa spúšťa cez okno *Login*, ktoré obsahuje *Runnable*. Po registrácii a prihlásení sa nachádzame v hlavnom menu kde máme na výber možnosti: *Testy*, *Profil*, *Ladderboard*, *Vytvor Test*.

5.1 Testy

Po kliknutí na tlačidlo testy sa nám zobrazí zoznam dostupných testov. Po vybratí testu si môžeme zvoliť možnosť časovača ktorý nastaví časomieru z odpočtom 30 sekúnd. Po kliknutí tlačidla Štart odštartujeme test kde odpovedáme na otázky rôzneho typu. Pri zadaní správnej odpovede a stlačenie tlačidla Ďalej sa zobrazí indikátor správnosti odpovede, pri nesprávnej odpovedi sa taktiež zobrazí správna. Po poslednej otázke sa test vyhodnotí a užívateľovi sa pripočítajú príslušné body. Pri splnení kritérií na získanie odznaku sa zobrazí nové okno informujúce o získaní príslušného odznaku.

5.2 Profil

V profile užívateľa sú zobrazené všetky údaje o užívateľovi vrátane odznakov a štatistík.

5.3 Ladderboard

Po kliknutí na Ladderboard sa zobrazí rebríček všetkých registrovaných užívateľov ktorý môžeme zoradovať podľa príslušných stĺpcov.

5.4 Vytvor test

Po kliknutí na Vytvor test môžeme zadať názov testu a následne pridať vhodný typ otázky, priradiť znenie a správnu odpoveď či viaceré odpovede a možnosti. Test sa následne na základe počtu otázok vytvorí, priradí každej otázke 1 bod a zobrazí sa v ponuke výberu testov.