

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

WEBOVÁ APLIKÁCIA PRE VÝUKU
PROGRAMOVACIEHO JAZYKA POMOCOU INÉHO
JAZYKA
BAKALÁRSKA PRÁCA

2021
MARTINA VESELÁ

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

WEBOVÁ APLIKÁCIA PRE VÝUKU
PROGRAMOVACIEHO JAZYKA POMOCOU INÉHO
JAZYKA
BAKALÁRSKA PRÁCA

Študijný program: aplikovaná informatika
Študijný odbor: informatika
Školiace pracovisko: FMFI.KAI - Katedra aplikovanej informatiky
Školiteľ: Ing. František Gyárfáš, PhD.

Bratislava, 2021
Martina Veselá



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Martina Veselá
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Webová aplikácia pre výuku programovacieho jazyka pomocou iného jazyka
Web application for learning of a programming language using another one

Anotácia: Cieľom bakalárskej práce je navrhnúť a vytvoriť interaktívne webové prostredie pre výuku programovania v programovacom jazyku (napr. Java) za pomoci znalostí iného programovacieho jazyka (napr. Python). Prostredie ponúkne študentovi riešenia úloh v jemu známom jazyku a vedie ho k preprogramovaniu týchto úloh do druhého jazyka. Úlohy budú definované formou testmi riadeného programovania (TDD). Aplikácia bude obsahovať editory kódu vo vybraných jazykoch a testovanie na serveri pomocou jednotkových testov. Systém bude realizovaný pomocou technológií/nástrojov: PHP alebo Python (framework), MySQL, HTML5, CSS, JavaScript (jQuery, Bootstrap), knižnice pre testovanie, virtuálny server.

Vedúci: Ing. František Gyarfaš, CSc.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 04.10.2020

Dátum schválenia: 04.01.2021

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Obsah

Úvod	1
1 Východiská	3
1.1 Potrebná teória	3
1.1.1 Webové prostredie	3
1.1.2 Editor kódu	4
1.1.3 Client-server	4
1.1.4 Unit testy	5
1.1.5 Test Driven Development	5
1.1.6 Virtual Machine	5
1.2 Existujúce riešenia	6
1.2.1 CodeAcademy	7
1.2.2 CodeWars	8
1.3 Použité technológie	9
1.3.1 Backend	9
1.3.2 Frontend	10

Zoznam obrázkov

1.1	Textový editor	4
1.2	Client-server model	4
1.3	Diagram procesu testmi riadeného programovania	5
1.4	Ukážka webového editora CodeAcademy	7
1.5	Ukážka webového editora CodeWars	8
1.6	Ukážka jednoduchého jednotkového testu v Python	9
1.7	Ukážka jednoduchého jednotkového testu v Jave	9
1.8	Demo ukážka vytvorenia CodeMirror editoru	11

Úvod

Cieľom tejto práce je navrhnúť a vytvoriť interaktívne webové prostredie pre výuku programovania v programovacom jazyku *Java*, za pomoci znalostí iného programovacieho jazyka - *Python*. Prostredie bude poskytovať úlohy, spolu s riešením v programovacom jazyku Python a návod na preprogramovanie daného problému do jazyka Java. Tieto úlohy budú definované formou testmi riadeného programovania (TDD - Test Driven Development). Aplikácia bude obsahovať editory kódu v daných jazykoch, spolu s testmi pre dané úlohy a výsledok po zbehnutí kódu. Testovanie bude prebiehať na serveri pomocou jednotkových testov.

Existuje veľa interaktívnych nástrojov na výuku programovacích jazykov, ale našla som žiaden, ktorý by využíval iný, používateľovi už známy jazyk na výuku pre neho nového jazyka. Výuka týmto spôsobom má mnohé výhody. Ak už používateľ ovláda základy programovania, väčšina nástrojov na výuku by sa mu ich snažila nanovo vysvetliť. Táto aplikácia sa sústreďuje na rozdiely v programovacích jazykoch, čiže naučí používateľa danú syntax.

Vďaka interaktívnemu elementu si používateľ môže všetko hneď vyskúšať a naučiť sa danú problematiku praxou. V niektorých zadaniach si môže meniť aj kód v oboch jazykoch a takisto zmeniť testy, ktoré budú zbiehané. Taktiež sa naučí písať testy a zažije si výhody (ale aj nevýhody) testmi riadeného programovania. Okrem úloh sa dá aplikácia využiť ako sandbox, umožňuje slobodu prepisovania kódu a testov v oboch jazykoch.

Kapitola 1

Východiská

V tejto kapitole sa bude nachádzať potrebná teória pre vytvorenie danej aplikácie. Taktiež tu budú ukážky podobných existujúcich riešení. Nakoniec bude vysvetlenie použitých technológií.

1.1 Potrebná teória

1.1.1 Webové prostredie

Webové prostredie je také, ku ktorému má používateľ prístup cez webový prehliadač. Vďaka tomu nemusí nič inštalovať a hneď po príchode na stránku má prístup k aplikácií. Z pohľadu developera vyvíjať webovú aplikáciu môže byť jednoduchšie, keďže existuje veľa knižníc a vývojových rámcov. Má to, ale svoje obmedzenia, každé webové prostredie používa HTML, CSS a JavaScript. Bez nich sa nedá zaobísť a nedajú sa nahradiť.

1.1.2 Editor kódu

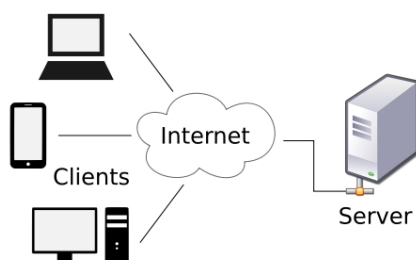
Editor kódu je špeciálny druh textového editora, ktorý je špecializovaný na písanie zdrojového kódu konkrétneho programovacieho jazyka. Dobrý editor používateľom prácu zjednodušuje pomocou zafarbenia, automatického odsadzovania, dopĺňaním koncových zátvoriek atď. Môže mať vlastné klávesové skratky a tiež môže byť schopný upozorniť používateľa na chybu. Taktiež môže ovládať syntax daného programovacieho jazyka. Väčšina má očíslované riadky pre jednoduchšiu orientáciu.

```
1 public class Main {  
2     public void test() {  
3         System.out.println("Tested.");  
4     }  
5  
6     public static void main(String[] args) {  
7         System.out.println("Ran.");  
8     }  
9 }  
10
```

Obr. 1.1: Textový editor - viď. rôznymi farbami zvýraznené typy kľúčových slov. V tejto ukážke je použitá populárna farebná téma Darkula

1.1.3 Client-server

Client-server model je model aplikačnej štruktúry, kde jeden alebo viac serverov poskytuje funkcionality, alebo servis jednému, alebo viacerým klientom. Väčšinou klient a server nezdieľajú spoločný hardvér, ale komunikujú medzi sebou na rôznych hardvéroch, a to buď cez domácu sieť alebo internet.



Obr. 1.2: Príklad client-server modelu

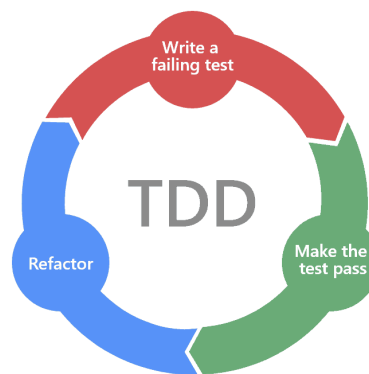
1.1.4 Unit testy

Jednotkové testy, bežne automatizované, kontrolujú funkčnosť kódu, a to po najmenších celkoch. Písanie testov nie je najpríjemnejšia činnosť, ale je veľmi užitočná na uistenie sa, že aplikácia funguje, ako má. Dá sa tak predchádzať chybám v okrajových prípadoch a pri zmenách kódu.

1.1.5 Test Driven Development

Testmi riadené programovanie (TDD) je metodológia programovania, pri ktorej sa počas celého developmentu využívajú testy. Postup programovania je nasledujúci. Najprv sa napíše test, ktorý musí zlyhať. Potom sa napíše metóda, ktorá ho splní a nepokazí doterajšie testy. Ak je to potrebné, zrefaktoruje sa. Potom sa proces opakuje, až pokým metóda nespĺňa dané požiadavky.

Táto metodológia je málokedy využívaná v praxi. Na začiatku projektu spomaľujú prácu a triviálne testy sa zdajú byť zbytočné.



Obr. 1.3: Diagram procesu testmi riadeného programovania

1.1.6 Virtual Machine

Virtuálna mašina je emulátor, ktorý používa softvér namiesto fyzického počítača. Umožňuje zbiehanie programov alebo aj celého operačného systému. Pri zbiehaní operačného systému môžete mať prístup k inému OS, než máte nainštalovaný, napr. pre developerské účely. Môže taktiež simulovať iný hardvér. Veľkou výhodou virtual mašiny je, že malvér alebo škodlivý zásah, ktorý sa dostane do virtual mašiny nemá prístup k danému počítaču.

1.2 Existujúce riešenia

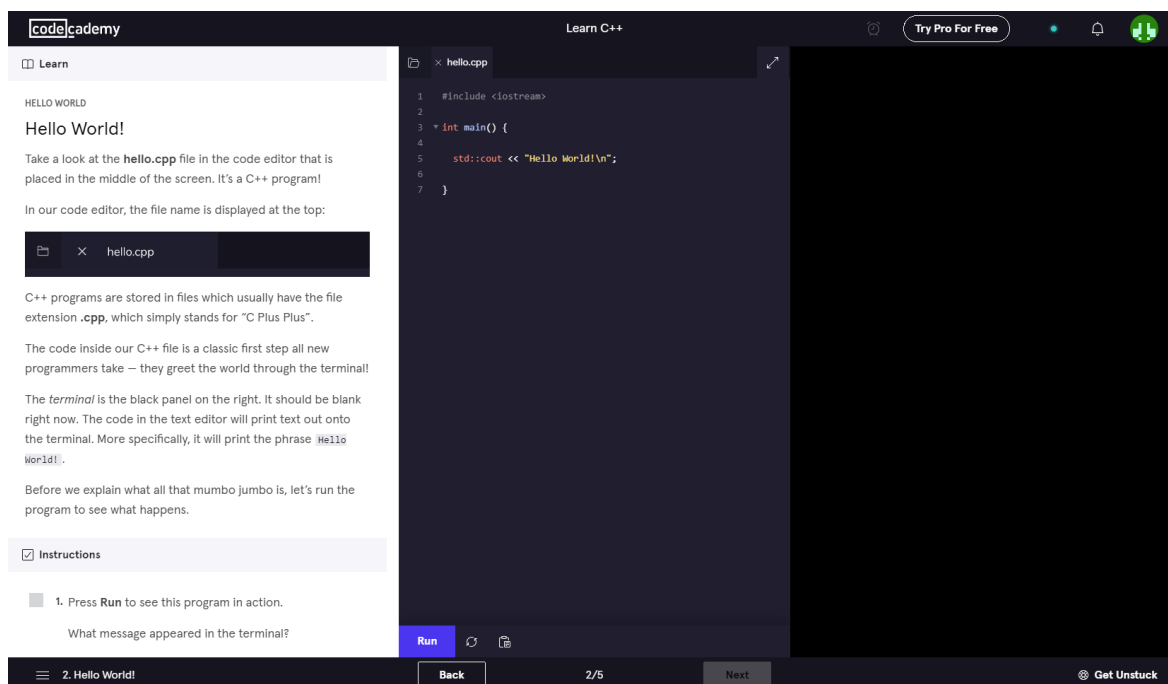
Nepodarilo sa mi nájsť žiadnu existujúcu aplikáciu, ktorá by učila programovací jazyk pomocou iného programovacieho jazyka.

Existuje veľmi veľa online kurzov vyučujúcich nejaký programovací jazyk. Väčšina takýchto stránok sa sústreďuje na výuku web developmentu. Takmer všetky stránky si vyžadujú registráciu používateľa. Pri niektorých je to veľmi jednoduchý a rýchly proces, iné si pýtajú veľa informácií o používateľovi.

Drvivá väčšina stránok má niektoré kurzy zadarmo (pováčšine začiatocnícke) a ostatné kurzy sa odomknú mesačným poplatkom.

Niektoré programovacie jazyky majú svoju vlastnú stránku na interaktívnu výuku. Príkladom takejto stránky by bol <https://tour.golang.org> pre programovací jazyk Go.

1.2.1 CodeAcademy



Obr. 1.4: Ukážka webového editora CodeAcademy

Codecademy je online interaktívna platforma, ktorá ponúka kurzy programovacích jazykov.

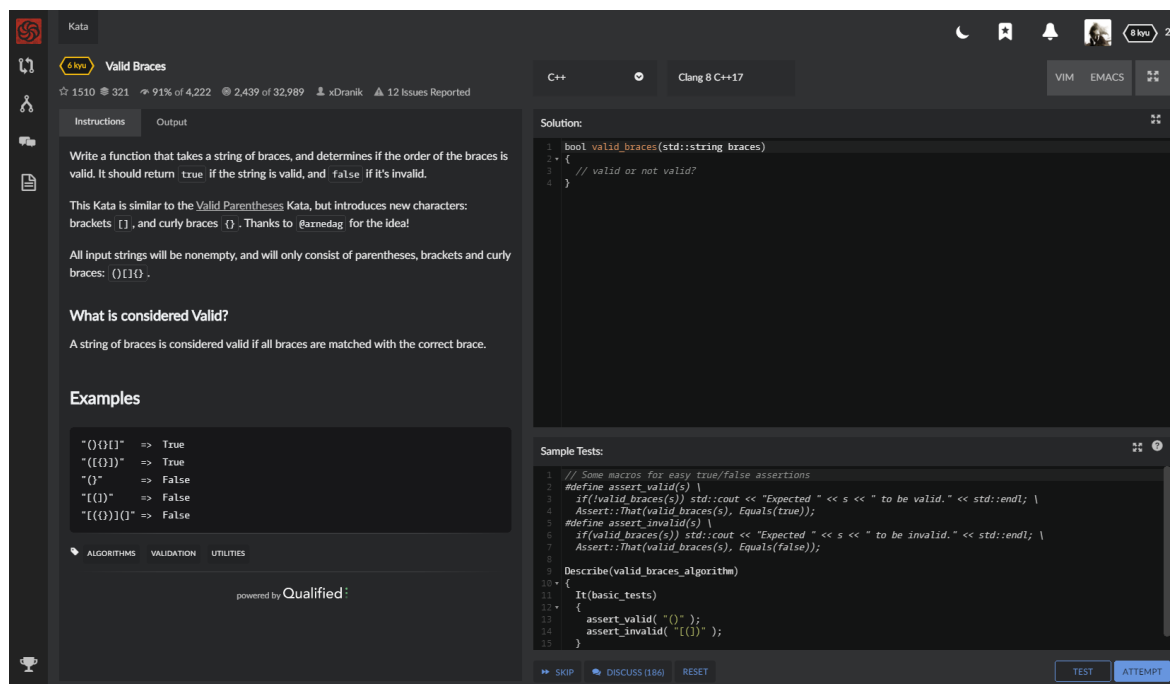
Registrácia je jednoduchá, dá sa pripojiť priamo cez účet na LinkedIn, Google, Facebook, Github alebo Twitter.

Momentálne ponúka kurzy v jazykoch HTML, CSS, Python, JavaScript, Java, SQL, Bash/Shell, Ruby, C++, R, C#, PHP, Go, Swift a Kotlin.

Na prístup ku všetkým kurzom treba platiť mesačný poplatok.

Na rozdiel od mojej aplikácie nevyužíva na výučbu ďalší programovací jazyk. Tak tiež sa nedá pozrieť si zbierané testy ani ich modifikovať.

1.2.2 CodeWars



Obr. 1.5: Ukážka webového editora CodeWars

CodeWars je tiež online interaktívna platforma s kurzmi programovacích jazykov.

Jej hlavnou výhodou je, že keď splníte zadanie, ukáže Vám riešenia ostatných používateľov, na ktorých sa dá naučiť, čo robiť lepšie alebo koľko je možných spôsobov implementácie.

Registrovať sa dá buď emailom a heslom alebo cez účet na GitHub.

Ponúkané kurzy sú v mnohých jazykoch: Clojure, CoffeeScript, C, Coq, C++, C#, Crystal, Dart, Elixir, F#, Go, Groovy, Haskell, Java, JavaScript, Kotlin, Lean, Lua, NASM, PHP, Python, Racket, Ruby, Rust, Scala, Shell, SQL, Swift a TypeScript.

Ďalej ponúka ďalšie jazyky, ktorých kurzy sú zatiaľ v bete: Agda, BF, CFML, COBOL, CommonLisp, Elm, Erlang, Haxe, Idris, Factor, Forth, Fortran, Julia, Nim, Objective-C, OCaml, Pascal, Perl, PowerShell, Prolog, PureScript, R, Raku, Reason, Solidity a VB.

Orientácia na stránke je trochu zložitá pre nového používateľa. Kurzy sú zadarmo, ale za mesačný poplatok sa odomknú prémiové funkcionality ako vyššie spomenuté porovnávanie riešení, výsledky testov ukázané priebežne, nie až po dobehnutí, beta funkcionality atď.

Síce neumožňuje výuku programacieho jazyka pomocou iného jazyka, ale testy sú viditeľné a dokonca aj editovateľné.

1.3 Použité technológie

1.3.1 Backend

Python

Aplikácia je programovaná v programovacom jazyku Python. Python je interpretovaný high-level programovací jazyk, ktorý je známy jednoduchosťou čítania kódu. Podľa mňa je to ideálny jazyk na rýchle, menšie projekty, ale pri veľkých dielach môže byť mäťúci, keďže premenné nemajú pevne zadáný typ. Backend mojej aplikácie nie je príliš rozsiahly, preto som sa rozhodla na neho použiť Python.

Python unittest

Vývojový rámec pre jednotkové testy v programovacom jazyku Python.

```
1  import unittest
2
3  class Test(unittest.TestCase):
4
5      def test(self):
6          self.assertEqual(1 + 1, 2)
7
8  if __name__ == '__main__':
9      unittest.main()
10
```

Obr. 1.6: Ukážka jednoduchého jednotkového testu v Python

JUnit

JUnit je vývojový rámec pre jednotkové testy v programovacom jazyku Java. Python unittest ním bol inšpirovaný.

```
1  import static org.junit.Assert.*;
2  import org.junit.Test;
3
4  public class Test {
5
6      @Test
7      public void test() {
8          assertEquals(1 + 1, 2);
9      }
10 }
```

Obr. 1.7: Ukážka jednoduchého jednotkového testu v Java

Docker

V tejto aplikácii sú používané jednotkové testy. Zbiehať cudzí kód nie je bezpečné, keďže by používateľ dostal prístup k počítaču, na ktorom beží server. Preto používam Docker, aby testy zbiehali virtuálne. Docker dostane text z editora aplikácie a pošle ho danému testeru (Python - UnitTest, Java - JUnit). Po zbehnutí testu sa výsledok pošle na stránku, kde používateľ môže vidieť, kde urobil chyby.

Docker je efektívnejší, než bežné virtuálne mašiny, keďže všetky jeho kontajnery zdieľajú servery jedného jadra operačného systému. Kontajnery sú navzájom izolované a obsahujú pre seba potrebný softvér, knižnice a konfiguračné súbory. Môžu medzi sebou komunikovať pomocou kanálov.

1.3.2 Frontend

TypeScript

TypeScript je open-source programovací jazyk od Microsoftu, nadstavba JavaScriptu. Pridáva statické typy, vďaka čomu je kód čitateľnejší, dokumentácia je lepšia a jeho funkcionality sa dá ľahšie kontrolovať. TypeScript podporuje ES6 moduly. ES6 - ECMAScript 2015 je JavaScriptový štandard, čo pomáha stránkam fungovať na rôznych webových prehliadačoch a pridáva užitočné funkcionality ako konštanty, šípkové funkcie, for cyklus, predvolené parametre a veľa ďalších.

React

React je open-source JavaScript knižnica pre stavbu používateľských rozhraní. Vie byť použitý ako báza pre jedno-stránkové webové stránky. Pre viac-stránkové stránky je potrebné použiť ďalšiu knižnicu na routing, ako napríklad React Router, ktorý aj používam v tomto projekte. Kód reaktu je tvorený pomocou komponentov. Keď sa komponent renderuje, dajú sa mu poslať hodnoty tzv. props. Komponenty môžu byť funkcie alebo triedy.

React Bootstrap

Najpopulárnejší frontend framework prerobený pre React. Rozloženie stránky sa robí veľmi jednoducho za pomoci Kontajnerov. Bootstrap obsahuje užitočné komponenty, ktoré sa dajú ľahko pridať na stránku ako tlačítka, dropdown menu, navigácia, tabuľky a mnoho ďalších. Netreba veľa úsilia na to, aby stránka vyzerala dobre.

CodeMirror

Versatilný textový editor implementovaný v JavaScripte špecializovaný na editovanie kódu. Všetky editory v mojej aplikácii sú pomocou neho vytvorené. Podporuje väčšinu programovacích jazykov. Jeho výzor sa dá prispôbiť a dá sa vybrať z populárnych farebných schém. Vie využívať klávesové skratky, ktoré sa dajú aj meniť. Automaticky odsadzuje riadky a zvýrazňuje zodpovedajúce zátvorky.

This is CodeMirror



Obr. 1.8: Demo ukážka vytvorenia CodeMirror editoru dostupná na ich stránke