

# WEB TEACHING TOOL FOR LEARNING AGILE PROGRAMMING

## WEBOVÁ VÝUKA AGILNÉHO PROGRAMOVANIA

Tamara Savkova

Školiteľ: Ing. František Gyárfáš, CSc.

# CIELE PRÁCE

- Návrh a implementácia interaktívneho webového prostredia pre výuku agilných metód programovania
- Metódy protirečia intuícii, ale pomáhajú písať čitateľný, udržiavateľný a rozšíriteľný kód
- Zoznámenie sa s teoretickou aj praktickou stránkou metód
- Znázornenie funkcionality pomocou ukázkových cvičení
- Pohodlná správa študijných materiálov

# EXISTUJÚCE RIEŠENIA

- Bakalárske a magisterské práce
  - Webová výuka programovania pomocou metodológie TDD.  
Tomáš Bočinec, 2018
  - Webová výuka programovania v C++ pomocou jednotkového testovania.  
Viliam Vakerman, 2017
  - Vývojové prostredie pre učenie sa refaktORIZÁCIE kódu.  
Ivan Latták, 2017
- Mass Open Online Courses
  - Codecademy
  - freeCodeCamp

# RIEŠENIE

- Modulárny systém
- Využitie moderných technológií
  - Front-end: Angular framework
  - Back-end: Spring framework
    - Správa Postgres databázy
    - Virtualizácia pomocou Docker kontajnerov
- Modul pre ohodnotenie riešenia (Java)

# RIEŠENIE

- Používateľské role: učiteľ a študent
- Výuka vo forme kurzov
- Automatické ohodnotenie riešení úloh
- Nástroje na správu materiálov kurzov
- Nástroje na riešenie úloh, verzionovanie

# ŠTRUKTÚRA KURZOV

Kurz

Cvičenie

Úlohy

## Course: Sample course

**Created:** Thursday, March 28, 2019. 11:08 GMT+01:00

**Description:** Web application functionality overview. Use agile programming methods to build a game, based on a legacy content

### Syllabus

Debugging a legacy program

Exercises on finding and fixing bugs in a legacy program



Intro

Theory

Solved

Finding the Bugs in Legacy Program

Black Box with Files

Not solved (0%)

Debugging the Legacy Program

Interactive Exercise with Files

Not solved (0%)

Manage exercises

# TYPY ÚLOH

Teória

Interaktívna úloha

Interaktívna úloha s  
textovými súbormi

Black Box

Black Box s  
textovými súbormi

# TYPY ÚLOH. TEÓRIA

## Obsah

- Popis

## Využitie

- Úvod
- Popis zdedeného riešenia
- Teoreticky materiál



# TYPY ÚLOH.

## INTERAKTÍVNA ÚLOHA

### Obsah

- Popis
- Editor pre zdrojový kód
- Editor pre testy

### Využitie

- Spúšťanie svojich testov na svojom zdrojovom kóde
- Spúšťanie učiteľských testov na svojom zdrojovom kóde

### Spätná väzba

- Výsledok kompilácie a testovania riešenia pomocou testov študenta
- Výsledok kompilácie a testovania riešenia pomocou testov učiteľa
- Pomer prejdenných testov k celkovému počtu

# TYPY ÚLOH.

## INTERAKTÍVNA ÚLOHA S TEXTOVÝMI SÚBORMI

### Obsah

- Popis
- Editor pre zdrojový kód
- Editor pre testy
- **Textový editor**

### Využitie

- Spúšťanie svojich testov na svojom zdrojovom kóde
- Spúšťanie učiteľských testov na svojom zdrojovom kóde
- **Testovanie**
- **Metódy práce so zdedeným kódom**

### Spätná väzba

- Výsledok kompilácie a testovania riešenia pomocou testov študenta
- Výsledok kompilácie a testovania riešenia pomocou testov učiteľa
- Pomer prejdenných testov k celkovému počtu

# TYPY ÚLOH. BLACK BOX

## Obsah

- Popis
- Editor pre testy

## Využitie

- Výučba TDD
- Výučba jednotkového testovania

## Spätná väzba

- Počet nájdenných chýb
- Celkový počet chýb
- Pomer nájdenných chýb k ich celkovému počtu

# TYPY ÚLOH. BLACK BOX S TEXTOVÝMI SÚBORMI

## Obsah

- Popis
- Editor pre testy
- **Textový editor**

## Využitie

- Výučba TDD
- Výučba jednotkového testovania
- **Testovanie**
- **Metódy práce so zdedeným kódom**

## Spätná väzba

- Počet nájdenných chýb
- Celkový počet chýb
- Pomer nájdenných chýb k ich celkovému počtu

# NÁSTROJE NA SPRÁVU MATERIÁLOV KURZOV

- Správa materiálov kurzov
  - Pridávanie, úprava, zrušenie, zmena poradia
- Dynamická formulár na vytváranie uloh
  - Zmena položiek formy v závislosti na zvolenom type úlohy
- Validácia položiek formulár
  - Je možné vytvoriť iba kompletnú úlohu
- Vytváranie dvoch verzii zdrojového kódu, testov a súborov
  - Skrytá verzia, ktorá je určená na testovanie riešenia
  - Verejná verzia pre študenta, ktorá sa zobrazí na začiatku riešenia úloh

# RIEŠENIE ÚLOH

- Stránka riešenia úlohy obsahuje
  - Navigačný panel pre prechádzanie medzi úlohami
  - Popis úlohy
  - Editory kódu pre zdrojový kód, testy a súbory
  - Tlačidlo pre odovzdanie riešenia
  - Panel pre výpis spätnej väzby

# Štítky pre súbory, na prechádzanie, premenovanie a mazanie

The screenshot shows an IDE interface. The top bar contains three tabs: 'Reversi.java', 'PlaygroundPrinter.java', and 'GameConfig.java'. The 'Reversi.java' tab is active, displaying the following code:

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 import java.nio.file.Files;
5 import java.nio.file.NoSuchFileException;
6 import java.nio.file.Path;
7 import java.util.ArrayList;
8 import java.util.Arrays;
9 import java.util.List;
10
11 public class Reversi {
12
13     int[][] playground;
14     private int leftB = 0;
15     private int leftW = 0;
16     private int[] players = new int[] { 1, 0 };
17     int onTurn = -1;
18     int winner = -1;
19     boolean ended = false;
20
21     Reversi() {
22     }
23
24     Reversi(Path gameFilePath) {
25         try {
26             String[] gameConfig = readGameConfig(gameFilePath);
27             int configFileLinesNumber = 3;
```

The sidebar on the right contains the following controls:

- Reset
- Load from solution
- Load from exercise:
- Debugging a legacy program
- Adding new features to the legacy program
- Refactoring the legacy program

Nastroje pre  
verzionovanie  
odovzdaných  
riešení

<

Reversi.java

✎

✕

PlaygroundPrinter.java

✎

✕

GameConfig.java

>

1

import java.io.BufferedReader;

2

import java.io.IOException;

3

import java.io.InputStreamReader;

4

import java.nio.file.Files;

5

import java.nio.file.NoSuchFileException;

6

import java.nio.file.Path;

7

import java.util.ArrayList;

8

import java.util.Arrays;

9

import java.util.List;

10

11

public class Reversi {

12

13

int[][] playground;

14

private int leftB = 0;

15

private int leftW = 0;

16

private int[] players = new int[] { 1, 0 };

17

int onTurn = -1;

18

int winner = -1;

19

boolean ended = false;

20

21

Reversi() {

22

}

23

24

Reversi(Path gameFilePath) {

25

try {

26

String[] gameConfig = readGameConfig(gameFilePath);

27

int configFileLinesNumber = 3;

Reset

Load from solution

Load from exercise:

Debugging a legacy program

Adding new features to the legacy program

Refactoring the legacy program

Obnovenie  
riešenia do  
pôvodného stavu

Načítanie  
odovzdaného  
riešenia úlohy

Načítanie  
odovzdaného  
riešenia  
ľubovoľnej úlohy  
v rámci kurzu



# UKÁŽKOVÝ KURZ

## CVIČENIE 1: DEBUGGING A LEGACY PROGRAM

### Úloha 1: Intro

- Typ: Teória
- Popis zdedeného riešenia, úvod do kurzu

### Úloha 2: Finding the Bugs in Legacy Program

- Typ: Black Box s textovými súbormi
- Hľadanie chýb v zdedenom riešení

### Úloha 3: Debugging the Legacy Program

- Typ: Interaktívna úloha s textovými súbormi
- Oprava nájdenných chýb

# UKÁŽKOVÝ KURZ

## CVIČENIE 2: ADDING NEW FEATURES TO THE LEGACY PROGRAM

### Úloha 1: Board Size

- Typ: Interaktívna úloha s textovými súbormi
- Bezpečné pridávanie novej funkcionality

### Úloha 2: Show Hints

- Typ: Interaktívna úloha s textovými súbormi
- Použitie agilnej metódy *sprout* na úpravu zdedeného kódu

# UKÁŽKOVÝ KURZ

## CVIČENIE 3: REFACTORING THE LEGACY PROGRAM

Úloha 1: Constants versus Enums

Úloha 2: No Duplication

Úloha 3: Do One Thing

Úloha 4: One Level of Abstraction per Function

Úloha 5: Small!

Úloha 6: Error Handling I

Úloha 7: Error Handling II

---

Typ: Interaktívna úloha  
s textovými súbormi

---

Každá úloha je  
venovaná jednotlivému  
kroku refaktORIZÁCII

# ZHRNUTIE

- Splnenie cieľov práce
- Další vývoj
  - Autentifikácia
  - Další moduly pre hodnotenie riešení

ĎAKUJEM ZA POZORNOST