



Тестова платформа

с
Java[™]

Клуб „Млад програмист“

Десислава Емилова Михайова и Звезделин Звезделинов Бончев
XI „А“ клас, СУ „Максим Райкович“, град Лясковец

23.06.2020

ЦЕЛИ

- ДА СЕ ПРОЕКТИРА ГРАФИЧЕН ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС С ВИЗУАЛНИ СРЕДСТВА
- ДА СЕ СЪЗДАДЕ ГРАФИЧНО ПРИЛОЖЕНИЕ С ИМЕ ТЕСТОВА ПЛАТФОРМА, ЧРЕЗ КОЯТО ДА СЕ СЪЗДАВАТ ТЕСТОВЕ ИЛИ РЕШАВАТ ТЕСТОВЕ.



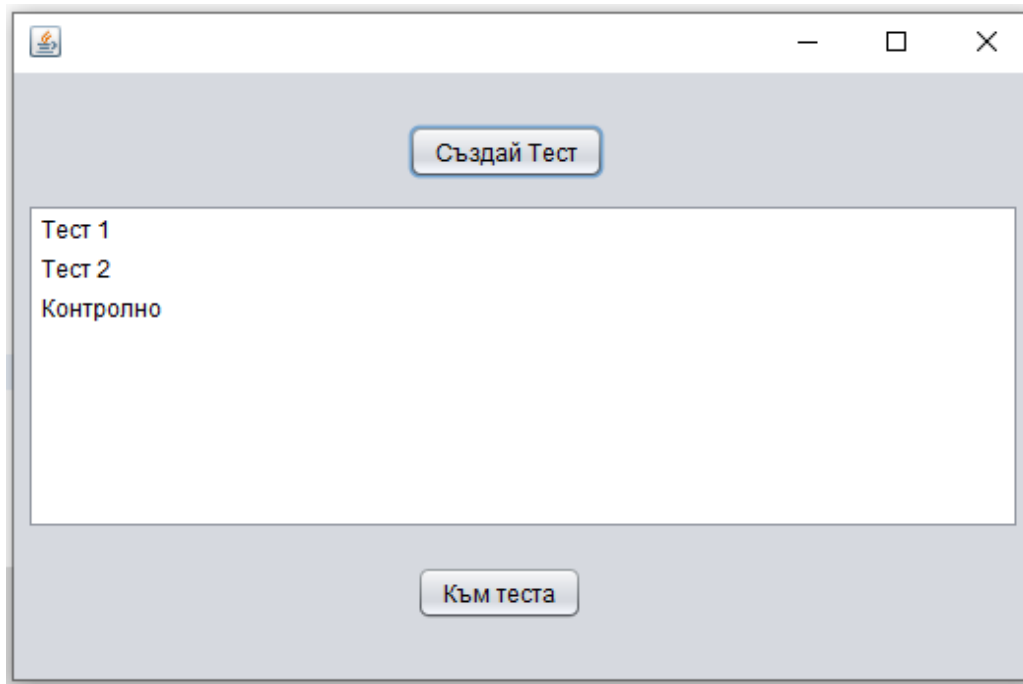
ГРАФИЧЕН ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС

Използват се:

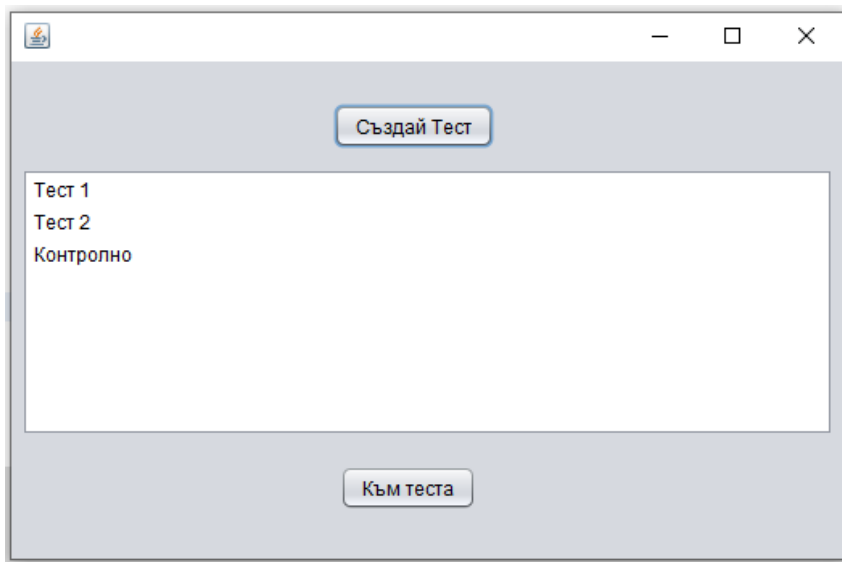
- Java Forms за потребителския интерфейс
- NetBeans за графичното му оформление



НАЧАЛЕН ЕКРАН



ДЕМОНСТРАЦИЯ НА ПРИЛОЖЕНИЕТО



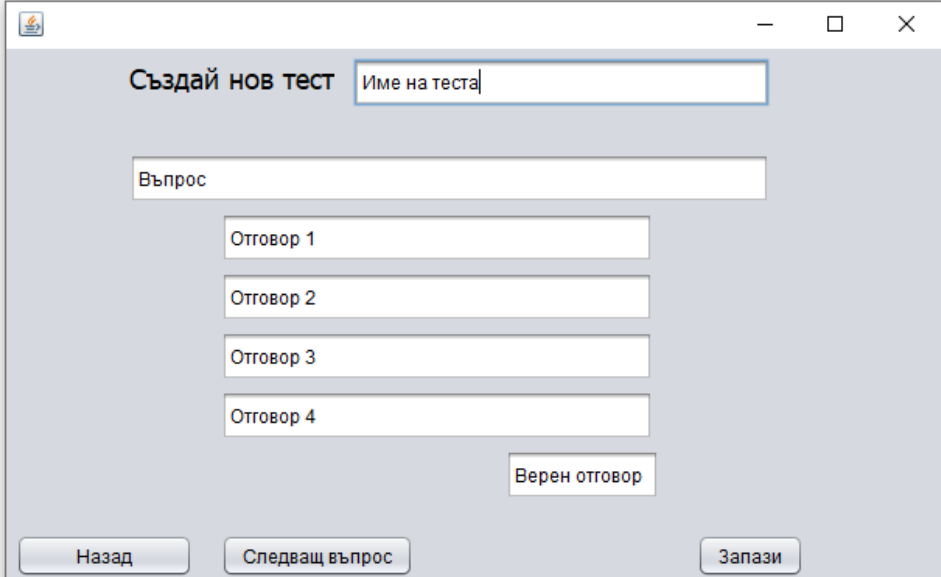
Можем или да създадем тест или при селектиране на един от съществуващите от списъка да преминем през теста



СЪЗДАВАНЕ НА ТЕСТ

Въвеждаме въпроса и 4-те възможни отговора, като в полето „верен отговор“ въвеждаме цифра от 1 до 4 за да посочим кой е верния отговор. При натискане на бутона „следващ въпрос“ запазваме въпроса и може да въведем нов.

При натискане на бутона „Запази“ запазваме целия тест и автоматично се връщаме на началното меню

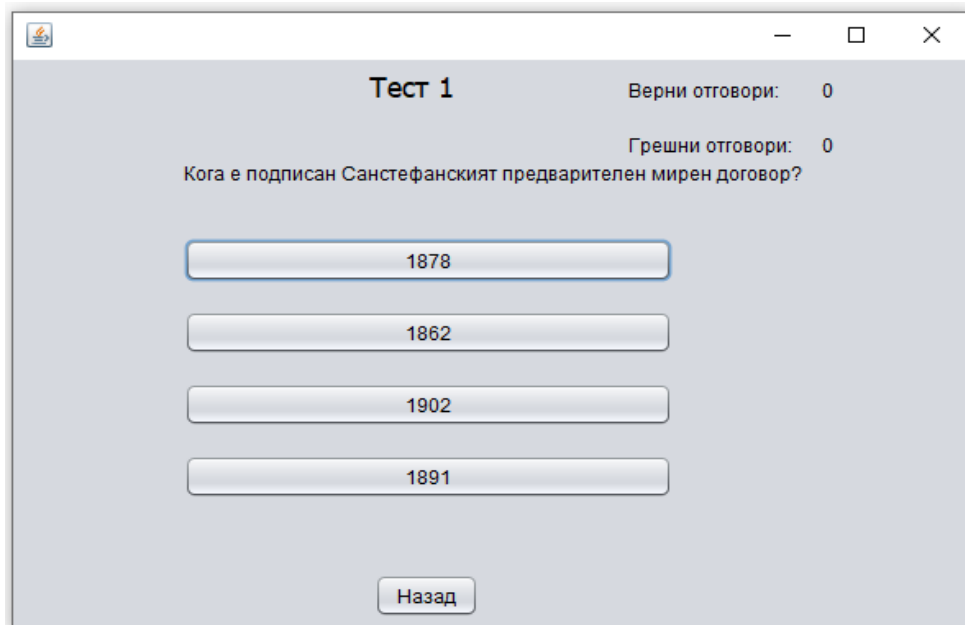


The screenshot shows a window titled "Създай нов тест" (Create new test). It contains a text input field for "Име на теста" (Test name). Below it is a larger text input field for "Въпрос" (Question). Underneath the question field are four smaller text input fields labeled "Отговор 1" (Answer 1), "Отговор 2" (Answer 2), "Отговор 3" (Answer 3), and "Отговор 4" (Answer 4). To the right of these fields is a dropdown menu currently showing "Верен отговор" (Correct answer). At the bottom of the window are three buttons: "Назад" (Back), "Следващ въпрос" (Next question), and "Запази" (Save).



ТЕСТЪТ

При натискане на бутона за отговора автоматично се отброява верен или грешен отговор в горната дясна част



Тест 1

Верни отговори: 0

Грешни отговори: 0

Кога е подписан Санстефанският предварителен мирен договор?

1878

1862

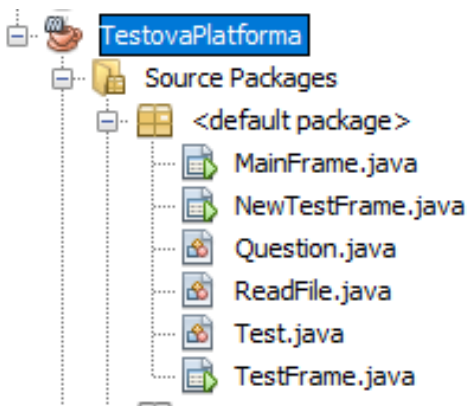
1902

1891

Назад



КОД НА ПРИЛОЖЕНИЕТО



Приложението се състои от общо 6 класа, като 3 от тях са форми за графичния потребителски интерфейс, а другите 3 са класове за въпросите и отговорите в приложението.

Форми:

- MainFrame.java – началното меню
- NewTestFrame.java – форма за създаване на нов тест
- TestFrame.java – форма за попълване на тест

Класове:

- Question.java – класът за въпроса
- ReadFile.java – класът за работа с текстовия файл
- Test.java – класът за теста



КОД НА ПРИЛОЖЕНИЕТО

(ReadFile.java)

С тази функция записваме нов тест във файла с тестовете. Редът #START бележи началото на нов тест. Непосредствено след него стои ред със заглавието на теста. Редът #QUESTION бележи началото на нов въпрос от теста. Следващите редове обозначават въпроса, верния отговор и 4те отговора един на нови редове.

```
public void WriteTest(Test t)
{
    try {
        Writer out = new BufferedWriter(new OutputStreamWriter(
            new FileOutputStream("tests.txt", true), "UTF8"));

        out.append("\r\n");
        out.append("#START\r\n");
        out.append(t.name + "\r\n");

        for (int i = 0; i < t.questions.size(); i++)
        {
            Question q = t.questions.get(i);

            out.append("#QUESTION" + "\r\n");
            out.append(q.question + "\r\n");
            out.append(String.valueOf(q.correctAnswer) + "\r\n");
            out.append(q.answer1 + "\r\n");
            out.append(q.answer2 + "\r\n");
            out.append(q.answer3 + "\r\n");
            out.append(q.answer4 + "\r\n");
        }

        out.flush();
        out.close();

    } catch (IOException e) {
        System.out.println("An error occurred.");
        e.printStackTrace();
    }
}
```



КОД НА ПРИЛОЖЕНИЕТО (ReadFile.java)

```
// Прочитане на файла с тестовете
public List<Test> ReadTests() throws IOException {
    try {
        List<Test> tests = new ArrayList<Test>();
        File file = new File("tests.txt");
        file.createNewFile();

        Test current = new Test("");
        BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(file), "UTF8"));

        String currentLine;
        while ((currentLine = br.readLine()) != null) {

            if (currentLine.equals("#START"))
            {
                String name = br.readLine();
                current = new Test(name);

                tests.add(current);
            }

            if (currentLine.equals("#QUESTION"))
            {
                String question = br.readLine();
                int correct = Integer.parseInt(br.readLine());

                String a1 = br.readLine();
                String a2 = br.readLine();
                String a3 = br.readLine();
                String a4 = br.readLine();

                Question q = new Question(question, a1, a2, a3, a4, correct);
                current.questions.add(q);
            }
        }

        br.close();

        return tests;
    } catch (FileNotFoundException e) {
        System.out.println("An error occurred.");
        e.printStackTrace();
    }
    return null;
}
```

С тази функция изчитаме файла с тестовете.

Следва се същата последователност от функцията за записване на файла.



КОД НА ПРИЛОЖЕНИЕТО

(TestFrame.java)

```
public void NextQuestion(int sequence)
{
    if (testFinished) return;

    if (sequence == correct)
        right++;
    else
        wrong++;

    correctCount.setText(String.valueOf(right));
    wrongCount.setText(String.valueOf(wrong));

    if (question == questionsCount) {
        FinishTest();
        return;
    }

    Question q = this.test.questions.get(question);
    correct = q.correctAnswer;
    questionLabel.setText(q.question);
    questionButton1.setText(q.answer1);
    questionButton2.setText(q.answer2);
    questionButton3.setText(q.answer3);
    questionButton4.setText(q.answer4);

    question++;
}
```

Това е функцията за преминаване от един въпрос на друг и зачисляването на верни и грешни отговори.



КОД НА ПРИЛОЖЕНИЕТО (MainFrame.java)

```
public MainFrame() {  
    initComponents();  
    ReadFile read = new ReadFile();  
  
    try {  
        tests = read.ReadTests();  
    } catch (IOException ex) {  
    }  
  
    DefaultListModel model = new DefaultListModel();  
    for (int i = 0; i < tests.size(); i++)  
    {  
        model.addElement(tests.get(i));  
    }  
    testList.setModel(model);  
}
```

Конструктора на основната форма MainFrame използва класът ReadFile за изчитане на файла и зареждането на тестовете в приложението.



КОД НА ПРИЛОЖЕНИЕТО

(MainFrame.java)

```
private void playTestButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    int selected = testList.getSelectedIndices()[0];  
    Test test = tests.get(selected);  
  
    this.dispose();  
    TestFrame frame = new TestFrame(test);  
    frame.setVisible(true);  
}
```

Функцията с която зареждаме селектирания тест и отваряме нова форма за попълване на теста (TestFrame.java)



ЗАКЛЮЧЕНИЕ И РАЗВИТИЕ

ТЕСТОВАТА ПЛАТФОРМА МОЖЕ ДА ИЗПОЛЗВА ПО ВСИЧКИ УЧЕБНИ ПРЕДМЕТИ ЗА ПОДГОТОВКА НА НВО, ДЗИ ИЛИ КОНТРОЛНИ И КЛАСНИ РАБОТИ.



БЛАГОДАРИМ ЗА ВНИМАНИЕТО!

