

Харьковский национальный университет им. В.Н. Каразина

Факультет компьютерных наук

Лабораторная работа №12

По учебной дисциплине

**Математические методы и технологии тестирования и верификации
программного обеспечения**

«Тестирование API»

Выполнил:

Студент группы КС-22

Филатов Виталий Витальевич

Проверил:

Доц. Малахов Сергей Витальевич

Тема: Тестирование API.

Цель работы: Изучить тестирование API, написать запросы к веб-ресурсу.

ВЫПОЛНЕНИЕ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

```
public class TestHttpConnection {

    @Test
    public void http_status_code_is_200() throws IOException {
        HttpURLConnection connection;
        String address = "http://localhost";
        connection = (HttpURLConnection) new URL(address).openConnection();
        connection.setRequestMethod("GET");
        connection.setUseCaches(false);
        connection.setConnectTimeout(2500);
        connection.setReadTimeout(2500);
        connection.connect();

        Assert.assertEquals(HttpURLConnection.HTTP_OK,
            connection.getResponseCode());
    }

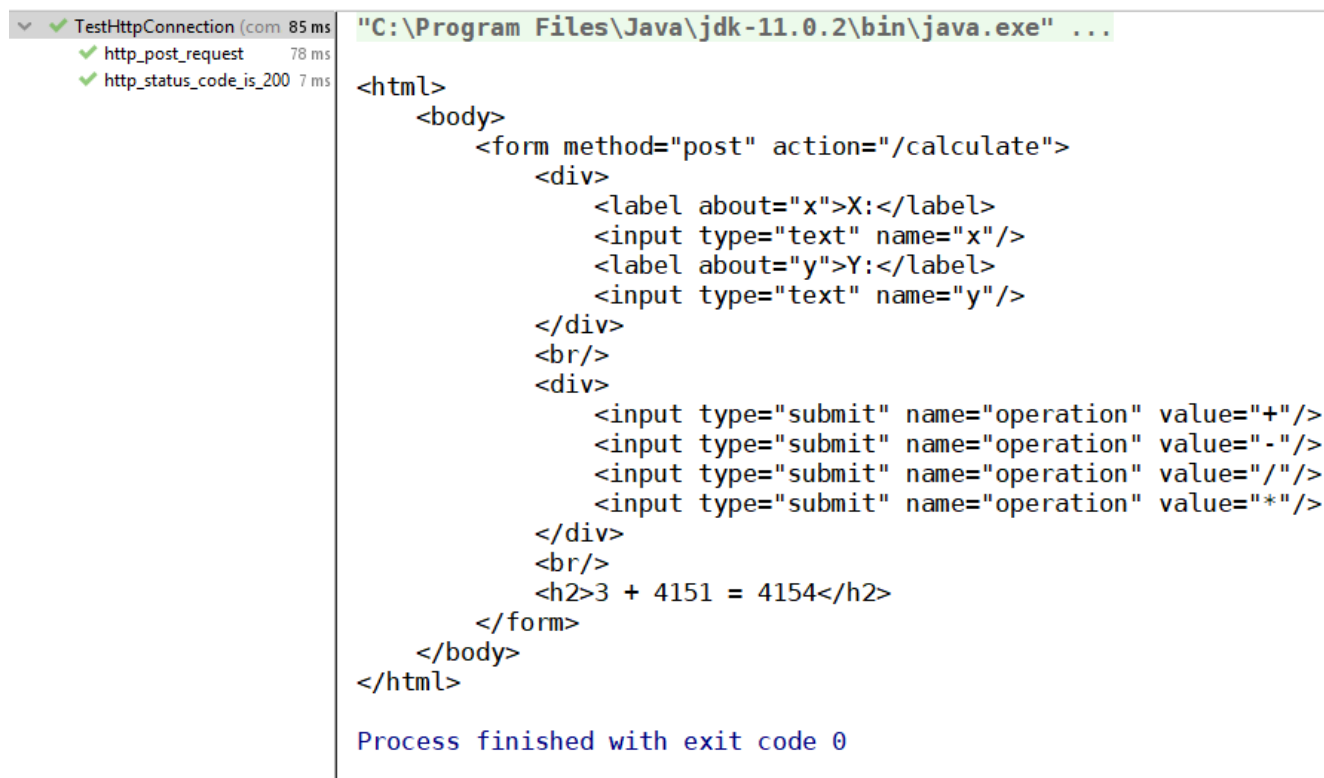
    @Test
    public void http_post_request() throws IOException {
        String parameters = "x=3&y=4151&operation=%2B";
        HttpURLConnection connection;
        String address = "http://localhost/calculate";
        connection = (HttpURLConnection) new URL(address).openConnection();
        connection.setRequestMethod("POST");
        connection.setUseCaches(false);
        connection.setConnectTimeout(2500);
        connection.setReadTimeout(2500);

        connection.setDoOutput(true);
        OutputStream outputStream = connection.getOutputStream();
        outputStream.write(parameters.getBytes());
        outputStream.flush();
        outputStream.close();

        InputStream inputStream = connection.getInputStream();
        BufferedReader reader = new BufferedReader(new
            InputStreamReader(inputStream));
        String line;

        while ((line = reader.readLine()) != null) {
            System.out.println(line);
        }

        Assert.assertEquals(HttpURLConnection.HTTP_OK,
            connection.getResponseCode());
    }
}
```



The screenshot shows an IDE window with a test runner on the left and a code editor on the right. The test runner shows three successful tests: 'TestHttpConnection' (85 ms), 'http_post_request' (78 ms), and 'http_status_code_is_200' (7 ms). The code editor displays an HTML document that represents a web form for calculations. The form has a method of 'post' and an action of '/calculate'. It contains two text input fields for 'x' and 'y', followed by four submit buttons for addition, subtraction, division, and multiplication. Below the form, the result of a calculation is displayed: '3 + 4151 = 4154'.

```

TestHttpConnection (com 85 ms
  ✓ http_post_request 78 ms
  ✓ http_status_code_is_200 7 ms

"C:\Program Files\Java\jdk-11.0.2\bin\java.exe" ...

<html>
  <body>
    <form method="post" action="/calculate">
      <div>
        <label about="x">X:</label>
        <input type="text" name="x"/>
        <label about="y">Y:</label>
        <input type="text" name="y"/>
      </div>
      <br/>
      <div>
        <input type="submit" name="operation" value="+"/>
        <input type="submit" name="operation" value="-"/>
        <input type="submit" name="operation" value="/"/>
        <input type="submit" name="operation" value="*/>
      </div>
      <br/>
      <h2>3 + 4151 = 4154</h2>
    </form>
  </body>
</html>

Process finished with exit code 0

```

Рисунок 1 – успешное прохождение тестов

ВЫВОДЫ

В ходе данной работы, я научился писать HTTP запросы (GET, POST). Были написаны два теста. Первый отправляет GET запрос на web приложение, разработанное для лабораторной работы №11 и проверяет что Status code равен 200. Второй тест отправляет POST запрос с параметрами «x=3», «y=4151», «operation=%2B». «%2B» соответствует символу «+». В результате отправки данного POST запроса нас перебрасывает на страницу, где виден результат сложения «x» и «y». Также проверяется чтобы Status code был 200.