



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение  
высшего образования*

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

---

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных  
технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3**  
**по дисциплине**  
**«Тестирование и верификация программного обеспечения»**

Выполнил студент группы ИКБО-01-20

Баикин К. Е.

Принял

Петренко А. А.

Практическая работа выполнена

«14» октября 2022г.

*(подпись студента)*

«Зачтено»

«  » \_\_\_\_\_ 2022 г.

*(подпись руководителя)*

Москва 2022

## СОДЕРЖАНИЕ

МЕТОДОЛОГИЯ РАЗРАБОТКИ ПО.....	3
ТЕМА И GIT.....	3
ПРИНЦИП РАБОТЫ.....	3
ДЕМОНСТРАЦИЯ РАБОТОСПОСОБНОСТИ .....	4
РУЧНОЕ ТЕСТИРОВАНИЕ.....	5
МОДУЛЬНЫЕ ТЕСТЫ .....	6
ИСХОДНЫЙ КОД.....	9
ВЫВОД.....	10

## **Методология разработки ПО**

Для разработки использована методология TDD (Test DriverDevelopment). Разработка через тестирование требует от разработчика создания автоматизированных модульных тестов, определяющих требования к коду непосредственно перед написанием самого кода. Тест содержит проверки условий, которые могут либо выполняться, либо нет. Когда они выполняются, говорят, что тест пройден. Прохождение теста подтверждает поведение, предполагаемое программистом.

## **Тема и GIT**

Для работы была выбрана тема «Бесследный уничтожитель файлов и папок». Гит-репозиторий: [https://github.com/kiruxa-projects/TESTING\\_PR3](https://github.com/kiruxa-projects/TESTING_PR3)

## **Принцип работы**

На вход падёт абсолютный путь к файлу или директории. Программа проверяет, существует ли объект по данному пути. Если нет, то в консоль выводится соответствующее сообщение. Если объект существует, то программа проверит, является ли объект директорией. Если да, то метод рекурсивно вызывает себя до тех пор, пока не будет найден файл или пустая директория. При нахождении файла его содержимое несколько раз перезаписывается. После этого файл удаляется. После удаления всех файлов удаляется директория.

## Демонстрация работоспособности

Для демонстрации работоспособности создадим файл testFile.txt. Также создадим директорию testDir с файлами testFile1.txt, testFile2.txt. (Рисунок 1 и Рисунок 2).

Имя	Дата изменения	Тип	Размер
.git	14.10.2022 20:35	Папка с файлами	
.idea	14.10.2022 20:30	Папка с файлами	
src	14.10.2022 20:30	Папка с файлами	
target	14.10.2022 20:30	Папка с файлами	
testDir	14.10.2022 20:43	Папка с файлами	
pom.xml	14.10.2022 20:30	Исходный файл ...	2 КБ
README.md	14.10.2022 20:30	Исходный файл ...	1 КБ
Практическая работа 2.docx	14.10.2022 20:42	Документ Micros...	57 КБ
testFile.txt	14.10.2022 20:43	Файл "TXT"	0 КБ

Рисунок 1 — Директория проекта

Имя	Дата изменения	Тип	Размер
testFile1.txt	14.10.2022 20:42	Файл "TXT"	0 КБ
testFile2.txt	14.10.2022 20:43	Файл "TXT"	0 КБ

Рисунок 2 — Папка testDir

## Ручное тестирование

Запустим нашу программу, укажем путь к файлу (Рис. 3).



Рисунок 3 — Запуск программы на удаление файла

Далее попробуем удалить папку (Рис. 4).



Рисунок 4 — Удаление директории

На рисунке 5 видно, что наша папка и файл удалились. Программа работает корректно.

Имя	Дата изменения	Тип	Размер
.git	14.10.2022 20:52	Папка с файлами	
.idea	14.10.2022 21:01	Папка с файлами	
src	14.10.2022 20:30	Папка с файлами	
target	14.10.2022 20:54	Папка с файлами	
pom.xml	14.10.2022 20:30	Исходный файл ...	2 КБ
README.md	14.10.2022 20:30	Исходный файл ...	1 КБ
Практическая работа 2.docx	14.10.2022 21:01	Документ Micros...	103 КБ

Рисунок 5 — Результат работы

## Модульные тесты

Для тестирования и разработки данной программы по методологии TDD были написаны следующие юнит тесты (Листинг 1).

### Листинг 1. Модульные тесты

```
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.mockito.junit.jupiter.MockitoExtension;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

import static org.junit.jupiter.api.Assertions.*;

@ExtendWith(MockitoExtension.class)
class CompleteDestructionTest {

    @Test
    void main() {
    }

    @Test
    void recursiveDelete() {
        File dir = new File("testDirectory");
        dir.mkdir();

        File file1 = new File(dir.getAbsolutePath()+"\\file1.txt");
        File file2 = new File(dir.getAbsolutePath()+"\\file2.txt");
        try {
```

```

        file1.createNewFile();
        file2.createNewFile();
    } catch (IOException e) {
        e.printStackTrace();
    }

    CompleteDestruction cm = new CompleteDestruction();
    cm.recursiveDelete(dir);

    Assertions.assertEquals(false, dir.exists());
}

@Test
void deleteFile() {
    File file = new File("Example.txt");

    CompleteDestruction cm = new CompleteDestruction();
    cm.deleteFile(file);

    Assertions.assertEquals(false, file.exists());
}

@Test
void overwriteFile() {
    // Создаём временный файл для тестирования
    try {
        File file = new File("Example.txt");

        // Создание файла
        file.createNewFile();

        CompleteDestruction cm = new CompleteDestruction();
        cm.overwriteFile(file, "НОВЫЕ СИМВОЛЫ");

        // Создание объекта FileReader
        String returnStr = "";
        FileReader fr = new FileReader(file);
        int c;
        while((c=fr.read())!=-1){
            returnStr = returnStr + (char)c;
        }
        fr.close();

        Assertions.assertEquals("НОВЫЕ СИМВОЛЫ", returnStr);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Результат прохождения всех тестов показан на рисунке 6.

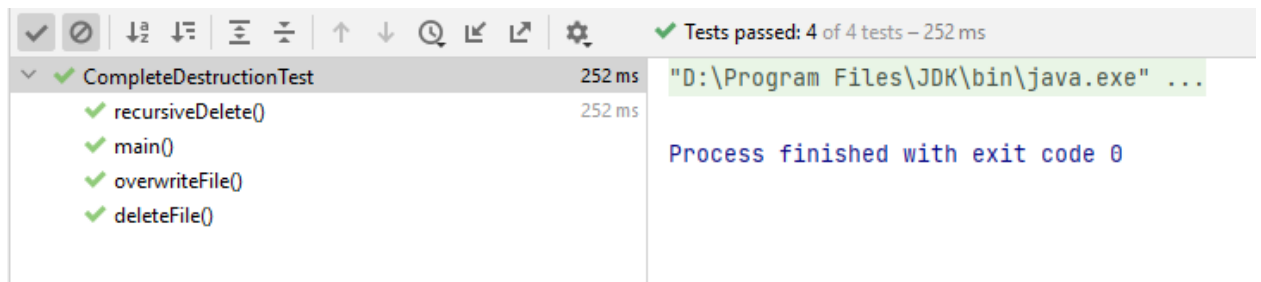


Рисунок 6 — Результат прохождения тестов



## Исходный код

Исходный код программы показан на листинге 2.

### Листинг 2. Исходный код программы

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.nio.file.Path;
import java.util.Scanner;

public class CompleteDestruction {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        String path = in.nextLine();

        File file = new File(path);
        if(!file.exists()){
            System.out.println("Неправильный путь");
            return;
        }

        recursiveDelete(file);
    }

    public static void recursiveDelete(File file){
        //если это папка, то идем внутрь этой папки и вызываем рекурсивное
удаление всего, что там есть
        if (file.isDirectory()) {
            for (File f : file.listFiles()) {
                recursiveDelete(f);
            }
        }

        if(file.isDirectory())
            file.delete();
        else {
            deleteFile(file);
        }
    }

    public static void deleteFile(File file){

        overwriteFile(file, "1516516565468");
        overwriteFile(file, "ergkjkregvrekgreg");
        overwriteFile(file, "weflew65f4ew6rv4r8e9vg46f");

        file.delete();
    }

    public static void overwriteFile(File file, String str){
        // Создание объекта FileWriter
        FileWriter writer = null;
        try {
            writer = new FileWriter(file);

            // Запись содержимого в файл
            writer.write(str);
            writer.flush();
            writer.close();
        }
    }
}
```

```
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

## **Вывод**

В ходе выполнения данной работы были укреплены навыки проектирования, разработки и тестирования программ. Были приобретены навыки работы с библиотекой Unit-тестирования. В ходе тестирования успешно выполнились все тесты, ручное тестирование выявило полную работоспособность программы в соответствии с поставленной задачей.

## **Список информационных источников**

1. Майерс Г., Баджетт Т., Сандлер К. Искусство тестирования программ. – М.: Вильямс, 2016. – 272 с.
2. Куликов С.А. Тестирование программного обеспечения. Базовый курс. Электронный ресурс. URL: [http://svyatoslav.biz/software\\_testing\\_book/](http://svyatoslav.biz/software_testing_book/) (последнее обращение 08.09.2022).
3. Котляров В.П. Основы тестирования программного обеспечения – Учебное пособие - М.: «ИНТУИТ», 2016 – 248 с., ЭБС Лань - <https://e.lanbook.com/book/100352#authors>.