

Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации
Ордена Трудового Красного Знамени
федеральное государственное бюджетное образовательное учреждение высшего образования
МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ

Кафедра «Математической кибернетики и информационных технологий»

Информационные технологии и программирование
Лабораторная работа №4

Выполнил: студент группы

БВТ2306

Кесслер Алексей Сергеевич

Москва, 2024 г.

Цель работы: Изучение использования исключений и их обработка

Задача работы: Создание собственных классов исключений.

Выполнение

Задание 1. Необходимо написать программу, которая будет находить среднее арифметическое элементов массива.

Для начала напишем метод, который может выбросить несколько видов исключений:

```
1 usage  DaNKAadnka
public static int avarageIntSum (Object[] arr, int size) {
    int av = 0;
    for (int i = 0; i < size; i++) {
        av += (int) arr[i];
    }
    return av;
}
```

Рисунок 1 - Метод avarageIntSum

И теперь в методе main мы напишем обработку этого исключения

```
Object[] a1 = {};
int n = a1.length;
try{
    System.out.println(avarageIntSum(a1, size: 0) / n);
}
catch(ClassCastException e) {
    System.out.println("There is no int array");
}
catch(ArrayIndexOutOfBoundsException f) {
    System.out.println("You out of your bounds");
}
catch(ArithmeticException g) {
    System.out.println("devided by zero");
}
```

Рисунок 2 - Обработка исключений

Задание 2. Необходимо написать программу, которая будет копировать содержимое одного файла в другой.

```

try {
    file1 = new File(path1);
    file2 = new File(path2);

    FileWriter f1 = new FileWriter(file2, append: true);

    Scanner scanner = new Scanner(file1);
    ArrayList<String> arr = new ArrayList<>();
    while (scanner.hasNextLine()) {
        arr.add(scanner.nextLine());
    }
    scanner.close();

    for (int i = 0; i < arr.size(); i++) {
        f1.append(arr.get(i));
        f1.append("\n");
    }
    f1.flush();
    //System.out.println(arr);
}
catch(FileNotFoundException e) {
    System.out.println("You should properly choose your files");
}
catch(IOException f) {
    System.out.println("Something wrong with your inout and output");
}
}

```

Рисунок 3 - Копирование содержимого одного файла в другой

Задание 3. Мой вариант - 5. Создайте класс CustomEmptyStackException, который будет использоваться для обработки исключения EmptyStackException.

Сначала напишем исключение, затем реализуем свой стек, который выбрасывает исключение.

Также напишем логирование исключения в текстовый файл.

```

public class CustomEmptyStackException extends Exception {

    3 usages
    private String message;

    no usages  DaNKAadnka
    public CustomEmptyStackException() { this.message = "Stack is empty"; }

    2 usages  DaNKAadnka
    public CustomEmptyStackException(String message){

        this.message = message;
    }

    DaNKAadnka
    public String getMessage() { return this.message; }
}

```

Рисунок 4 - Кастомное исключение для стека

```

public class CustomStack <T> {

    6 usages
    private int size;
    4 usages
    Stack<T> ourStack;

    1 usage  DaNKAadnka
    public CustomStack () {
        ourStack = new Stack<T>();
        size = 0;
    }

    1 usage  DaNKAadnka
    public void pop () throws CustomEmptyStackException {
        if (size == 0) {
            try (FileWriter f1 = new FileWriter( fileName: "C://Users//79085//IdeaProjects//Exc
                f1.write( str: "You can't remove item from stack\n");
            }
            catch (IOException e) {
                //
            }
            throw new CustomEmptyStackException("You can't remove item from stack");
        }
        else {
            ourStack.pop();
            size -= 1;
        }
    }
}

```

Рисунок 5 - Кастомный стек

```

public void push (T a) {
    ourStack.push(a);
    size += 1;
}

1 usage  DaNKAadnka
public T get () throws CustomEmptyStackException {
    if (size == 0) {
        try (FileWriter f1 = new FileWriter( fileName: "C://Users//79085//IdeaProjects//Excepti
            f1.write( str: "You have nothing in your own stack\n");
        }
        catch (IOException e) {
            //
        }
        throw new CustomEmptyStackException("You have nothing in your own stack");
    }
    else {
        return ourStack.peek();
    }
}

no usages  DaNKAadnka
public int size() {
    return this.size;
}

```

Рисунок 6 - Кастомный стек(продолжение)

Репозиторий: <https://github.com/LightninG8/MTUCI/tree/main/%D0%98%D0%A2%D0%98%D0%9F/%D0%9B%D0%B0%D0%B1%D1%8B/%D0%9B%D0%B0%D0%B1%D0%B0%204>