**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа №4

По дисциплине «Информационные технологии и программирование»

Выполнил: Студент группы

БВТ2203

Попов Александр

Москва

2023

**Цель работы:** Знакомство с исключениями в ЯП Java.

**Задание 1:** Необходимо написать программу, которая будет находить среднее арифметическое элементов массива. При этом программа должна обрабатывать ошибки, связанные с выходом за границы массива и неверными данными (например, если элемент массива не является числом)

**Задание 2:** Необходимо написать программу, которая будет копировать содержимое одного файла в другой. При этом программа должна обрабатывать возможные ошибки, связанные с: Чтением и записью файлов (2в).

**Задание 3:** Создайте Java-проект для работы с исключениями. *Вариант 8*: Создайте класс CustomUnsupportedOperationException, который будет использоваться для обработки исключения UnsupportedOperationException. Реализуйте программу, которая выполняет математические операции (сложение, вычитание, умножение, деление) с помощью собственного класса и выбрасывайте исключение CustomUnsupportedOperationException, если операция не поддерживается.

**Ход работы**

**Задача 1:**

public *class* ArrayAvg {

*public* *static* int elementAvg(int[] *arr*) {

        int sum = 0;

*if* (*arr*.length == 0) {

*return* sum;

        }

*try* {

*for* (int i=0; i<*arr*.length+1; i++) {

                sum += *arr*[i];

            }

        } *catch* (ArrayIndexOutOfBoundsException *error*) {

            System.out.println("");

*return* 0;

        } *catch* (NumberFormatException *error*) {

            System.out.println("Invalid input");

*return* 0;

        }

*return* sum / *arr*.length;

    }

*public* *static* void main(String[] *args*) {

        System.out.println(elementAvg(*new* int[]{1, 2, 3, 4, 5}));

    }

}

Файл 1. ArrayAvg.java.

Я создал метод elementAvg, который считает среднее арифметическое и обрабатывает исключения, которые могут возникнуть при выходе за рамки массива и невалидный ввод данных пользователем.

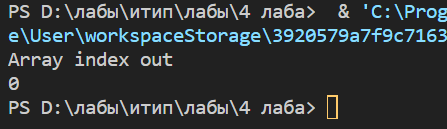
****

Рис. 1. Результат выполнения программы.

**Задача 2:**

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.FileInputStream;

*public* *class* CopyFile {

*public* *static* void copyContent(File *a*, File *b*) throws Exception {

        FileInputStream in = *new* FileInputStream(*a*);

        FileOutputStream out = *new* FileOutputStream(*b*);

*try* {

            int n;

*while* ((n = in.read()) != -1) {

                out.write(n);

            }

        } *catch* (IOException *e*) {

            System.out.println("Player file wouldn't load!");

*return*;

        } *finally* {

*if* (in != null) {

                in.close();

            }

*if* (out != null) {

                out.close();

            }

        }

        System.out.println("File Copied");

    }

*public* *static* void main(String[] *args*) throws Exception {

        File x = *new* File("4лаба.docx");

        File y = *new* File("new.docx");

        copyContent(x, y);

    }

}

Файл 2. CopyFile.java.

Здесь я создал метод CopyContent, который принимает файлы a и b и копирует содержимое a в b, в случае, если файловая система не поддерживается компьтером, будет вызвано исключение. Предварительно я создал 2 .docx файла, один заполненный, а другой новый, полностью пустой. Результатом выполнения программы будет заполнение 2 файла.

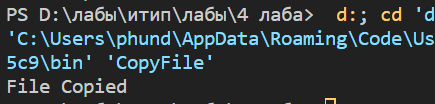


Рис. 2. Результат выполнения программы.

**Задача 3:**

public *class* CustomUnsupportedOperationException extends Exception {

*public* CustomUnsupportedOperationException(String *message*) {

        super(*message*);

    }

}

Файл 3. CustomUnsupportedOperationException.java

public *class* MyCalc {

*public* *static* double operation(String *op*, double *a*, double *b*) throws CustomUnsupportedOperationException {

*switch* (*op*) {

*case* "+"*:*

*return* *a*+*b*;

*case* "-"*:*

*return* *a*-*b*;

*case* "\*"*:*

*return* *a*\**b*;

*case* "/"*:*

*try* {

*return* *a*/*b*;

                } *catch* (ArithmeticException *e*) {

                    System.out.println("Division by zero!");

*return* 0;

                }

*default:*

*throw* *new* CustomUnsupportedOperationException("Invalid operation!");

        }

    }

}

Файл 4. MyCalc.java

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

*public* *class* Main {

*public* *static* void main(String[] *args*) {

        double res1, res2;

*try*{

            res1 = MyCalc.operation("/", 2, 3);

            System.out.println(res1);

            res2 = MyCalc.operation("&", 4, 1);

            System.out.println(res2);

        } *catch* (CustomUnsupportedOperationException *e*) {

            logException(e);

            System.out.println(e);

        }

    }

*public* *static* void logException(Exception *e*) {

*try* (PrintWriter writer = *new* PrintWriter(*new* FileWriter("error.log", true))) {

*e*.printStackTrace(writer);

        } *catch* (IOException *exception*) {

            System.out.println("Error: " + exception);

        }

    }

}

Файл 5. Main.java.

Для начала я создал класс – исключение CustomUnsupportedOperationException. Затем написал класс калькулятора с единственным методом operation. Внутри него, в случае, если пользователь передал несуществующую операцию, будет вызвано мое исключение.

Также я написал метод logException для логирования ошибок.

Затем в методе main я проверил работу программы.

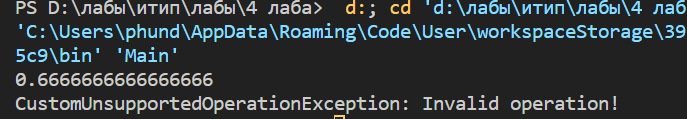


Рис. 3. Результат выполнения программы.

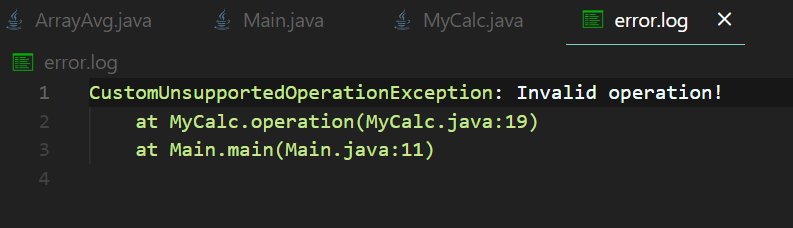


Рис. 4. Логирование ошибки.

**Выводы**

1. Я научился работать с исключениями, обрабатывать их и создавать свои собственные.