

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 «Математическая кибернетика и информационные технологии»

Отчёт по лабораторной работе № 4

по дисциплине «Информационные технологии и программирование»

Выполнил: студент группы БПИ2404 Журавлев Р. В.

Руководитель: Рыбаков Е. Д.

Москва 2025

Цель работы: Изучить работу исключений, научиться их обрабатывать и создавать собственные классы исключений в языке программирования Java.

Задание 1: 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).

```
public class ArrayAverage {
    public static void main(String[] args) {

        Object[] arr = {"Mouse", 92, 3.15, 18, 84.3, "word", 31, "OBJECT"};
        // Object[] arr = null;

        double sum = 0;
        int count = 0;

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

                if (arr[i] instanceof Number) {
                    sum += ((Number) arr[i]).doubleValue();
                    count++;
                } else {
                    System.out.println("Элемент с индексом " + i + " не
является числом.");
                }
            }

            double average = sum / count;
            System.out.println("Среднее арифметическое массива: " + average);

        } catch (ArrayIndexOutOfBoundsException ex) {
            System.out.println("Ошибка: выход за границы массива.");
        } catch (NullPointerException ex) {
            System.out.println("Ошибка: массив не инициализирован.");
        }
    }
}
```

Задание 2: Написать программу, которая будет копировать содержимое одного файла в другой. При этом программа должна обрабатывать возможные ошибки, связанные:

- с открытием и закрытием файлов (вариант 1);
- чтением и записью файлов (вариант 2).

```
import java.io.IOException;
import java.nio.file.*;

public class FileCopy {
    public static void main(String[] args) {

        Path sourceFile = Paths.get("Labs\\Lab4\\task2\\source.txt");
        Path destinationFile =
Paths.get("Labs\\Lab4\\task2\\destination.txt");

        try {
            Files.copy(sourceFile, destinationFile,
StandardCopyOption.REPLACE_EXISTING);
            System.out.println("Содержимое файла скопировано.");

        } catch (java.nio.file.NoSuchFileException ex) {
            System.out.println("Ошибка открытия: файл не существует - " +
ex.getMessage());
        } catch (java.nio.file.AccessDeniedException ex) {
            System.out.println("Ошибка: нет прав на изменение файла - " +
ex.getMessage());
        } catch (IOException ex) {
            System.out.println("Ошибка ввода / вывода при чтении / записи: " +
ex.getMessage());
        }
    }
}
```

Задание 3: Создайте Java-проект для работы с исключениями. Для каждой из указанных ниже задач напишите собственный класс для обработки исключений. *Создайте обработчик исключений, который логирует информацию о каждом выброшенном исключении в текстовый файл.*

Создайте класс CustomAgeException, который будет использоваться для обработки недопустимых возрастов. Реализуйте программу, которая проверяет возраст пользователя с использованием этого класса, и, если возраст меньше 0 или больше 120, выбрасывайте исключение CustomAgeException.

Создайте класс CustomEmailFormatException, который будет использоваться для обработки недопустимого формата email-адреса. Реализуйте программу, которая проверяет формат email-адреса с использованием этого класса, и, если адрес не соответствует формату, выбрасывайте исключение CustomEmailFormatException.

```
public class CustomAgeException extends Exception {  
    public CustomAgeException(String message) {  
        super(message);  
    }  
}
```

```
public class AgeValidator {  
    public static void validateAge(int age) throws CustomAgeException {  
        if (age < 0 || age > 120) {  
            throw new CustomAgeException("Недопустимый возраст (" + age + ").  
Возраст должен быть 0 до 120 лет.");  
        } else {  
            System.out.println("Возраст " + age + " допустим.");  
        }  
    }  
}
```

```
public class CustomEmailFormatException extends Exception {  
    public CustomEmailFormatException(String message) {  
        super(message);  
    }  
}
```

```
import java.util.regex.Pattern;
```

```
public class EmailValidator {
    private static final String emailRegex =
"^[A-Za-z0-9+_.-]+@[A-Za-z0-9.-]+\.\w{2,}\$";
    private static final Pattern pattern = Pattern.compile(emailRegex);

    public static void validateEmail(String email) throws
CustomEmailFormatException {
        if (email == null || !pattern.matcher(email).matches()) {
            throw new CustomEmailFormatException("Email-адрес (" + email + ")"
имеет некорректный формат.");
        } else {
            System.out.println("Email-адрес " + email + " допустим.");
        }
    }
}
```

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.time.LocalDateTime;

public class ExceptionLogger {
    private static final String logFile =
"Labs\\Lab4\\task3\\exceptions_logs.txt";

    public static void logException(Exception exception) {

        try (FileWriter fileWriter = new FileWriter(logFile, true);
PrintWriter writer = new PrintWriter(fileWriter)) {

            String timestamp = LocalDateTime.now().toString();
            String exceptionType = exception.getClass().getName();

            writer.println("[" + timestamp + "] " + exceptionType + ": " +
exception.getMessage());

        } catch (IOException ex) {
            System.err.println("Ошибка при записи в лог-файл: " +
ex.getMessage());
        }
    }
}
```

```
public class Main {
    public static void main(String[] args) {
```

```
int[] ages = {52, -1, 121};

String[] emails = {"bibal2@gmail.com", "das@d@", "EEE",
"boba13@yandex.ru"};

for (int age : ages) {
    try {
        AgeValidator.validateAge(age);
    } catch (CustomAgeException ex) {
        System.out.println(ex.getMessage());
        ExceptionLogger.logException(ex);
    }
}

for (String email : emails) {
    try {
        EmailValidator.validateEmail(email);

    } catch (CustomEmailFormatException ex) {
        System.out.println(ex.getMessage());
        ExceptionLogger.logException(ex);
    }
}
}
```

Вывод: В ходе лабораторной работы был изучен принцип работы исключений в языке программирования Java.