**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное учреждение высшего образования

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

**ФАКУЛЬТЕТ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**

Кафедра «МКиИТ»

**Информационные технологии и**

**программирование**

**Лабораторная работа по ИТП №4**

Выполнил: студент группы БВТ2205

Белов Егор Олегович

**Задание 1:**

public class ArrayAverage {  
 public static void main(String[] args) {  
 try {  
 // Ошибка NumberFormatException:  
 //int[] arr = {1, 2, Integer.parseInt("s"), 4, 5};  
  
 int[] arr = {1, 2, 3, 4, 5};  
  
 int sum = 0;  
  
 int steps = 0;  
  
 /\* Ошибка ArrayIndexOutOfBoundsException:  
 for (int i = 0; i <= arr.length; i++) {  
 sum += arr[i];  
 steps += 1;  
 }\*/  
  
 for (int element : arr) {  
 sum += element;  
 steps += 1;  
 }  
  
 System.*out*.println((double) sum / (double) steps);  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println("Вышел за массив.");  
 } catch (NumberFormatException e) {  
 System.*out*.println("Элемент не может быть преобразован в число.");  
 }  
 }  
}

В этом коде создан класс ArrayAverage, который содержит метод main. В методе main выполняется следующая последовательность действий:

Создается массив arr, содержащий целочисленные значения {1, 2, 3, 4, 5}.

Инициализируются переменные sum и steps со значениями 0. sum используется для сохранения суммы элементов массива arr, а steps - для подсчета количества элементов в массиве.

В цикле for-each происходит итерация по каждому элементу массива arr. Каждый элемент прибавляется к переменной sum. Значение steps увеличивается на 1.

Выводится результат деления суммы элементов sum на количество элементов steps, приведенный к типу double.

В случае возникновения исключения ArrayIndexOutOfBoundsException (выход за пределы массива) в блоке catch, будет выведено сообщение "Вышел за массив.". В случае возникновения исключения NumberFormatException (неправильный формат числа) в блоке catch, будет выведено сообщение "Элемент не может быть преобразован в число.".

Закомментированные строки содержат код, который вызывает ошибки исключений NumberFormatException и ArrayIndexOutOfBoundsException

**Задание 2 (Вариант: 2):**

import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
public class FileCopy {  
 public static void main(String[] args) {  
 String sourceFilePath = "src/firstFile.txt";  
 String destinationFilePath = "src/secondFile.txt";  
  
  
 try {  
 FileInputStream inputFile = new FileInputStream(sourceFilePath);  
 FileOutputStream outputFile = new FileOutputStream(destinationFilePath);  
  
 byte[] buffer = new byte[1024];  
 int bytesRead;  
  
 while((bytesRead = inputFile.read(buffer)) != -1) {  
 outputFile.write(buffer, 0, bytesRead);  
 }  
  
 String endComment = "\nСкопированный текст :3";  
 byte[] endCommentBytes = endComment.getBytes();  
 outputFile.write(endCommentBytes, 0, endCommentBytes.length);  
  
 System.*out*.println("Данные файла успешно скопированы.");  
  
 inputFile.close();  
 outputFile.close();  
 } catch (FileNotFoundException e) {  
 System.*out*.println("Неправильные пути к файлу.");  
 } catch (IOException e) {  
 System.*out*.println("Ошибка при чтении или записи файла.");  
 }  
 }  
}

В данном коде создан класс FileCopy, который содержит метод main. В методе main выполняется следующая последовательность действий:

Задаются строки sourceFilePath и destinationFilePath, содержащие пути к файлам для копирования и результирующему файлу соответственно.

В блоке try открывается FileInputStream для чтения данных из исходного файла sourceFilePath.

В блоке try открывается FileOutputStream для записи данных в целевой файл destinationFilePath.

Создается буфер типа byte[] размером 1024 байта.

Выполняется цикл, в котором данные считываются из исходного файла в буфер и записываются в целевой файл до тех пор, пока не будет достигнут конец файла (inputFile.read(buffer) возвращает -1).

После окончания копирования текстовая строка endComment преобразуется в массив байт (endCommentBytes) и записывается в целевой файл.

Выводится сообщение "Данные файла успешно скопированы."

Исходный файл и целевой файл закрываются методами close() для освобождения ресурсов.

Если возникает исключение FileNotFoundException, будет выведено сообщение "Неправильные пути к файлу."

Если возникает исключение IOException, будет выведено сообщение "Ошибка при чтении или записи файла."

**Задание 3 (Вариант: 2):**

import java.io.\*;  
  
public class Program {  
 public static void main(String[] args) {  
 String path = "src/errorList.txt";  
  
 try {  
 *age*(0, path);  
 *age*(120, path);  
 //age(-1, path);  
 //age(125, path);  
 } catch (CustomAgeException e) {  
 System.*out*.println("Значение не удовлетворяет условию.");  
 } catch (FileNotFoundException e) {  
 System.*out*.println("Не удалось найти файл.");  
 } catch (IOException e) {  
 System.*out*.println("Ошибка при чтении или записи файла.");  
 }  
 }  
  
 public static void age(int ageOfPerson, String path) throws CustomAgeException, IOException {  
 if (ageOfPerson < 0) {  
 throw new CustomAgeException(ageOfPerson, path);  
 } else if (ageOfPerson > 120) {  
 throw new CustomAgeException(ageOfPerson, path);  
 }  
 System.*out*.println(ageOfPerson);  
 }  
}  
  
class CustomAgeException extends Exception {  
 CustomAgeException(int age, String path) throws IOException {  
 BufferedWriter writer = new BufferedWriter(new FileWriter(path, true));  
  
 if (age < 0) {  
 System.*out*.println(age + " < " + 0);  
 writer.write(age + " < " + 0);  
 } else if (age > 120) {  
 System.*out*.println(age + " > " + 0);  
 writer.write(age + " > " + 0);  
 }  
  
 writer.newLine();  
 writer.close();  
 }  
}

**В этом коде определен класс Program, который содержит метод main() в качестве точки входа программы. Внутри метода main() определена переменная path, которая содержит путь к файлу "src/errorList.txt".**

**Затем в блоке try вызывается метод age() с различными значениями параметра ageOfPerson и переданным путем к файлу path. Метод age() проверяет значение параметра ageOfPerson и выбрасывает исключение CustomAgeException, если значение не удовлетворяет условиям (меньше 0 или больше 120).**

**В блоке catch обрабатываются возможные исключения. Если возникло исключение CustomAgeException, выводится сообщение "Значение не удовлетворяет условию.". Если возникло исключение FileNotFoundException, выводится сообщение "Не удалось найти файл.". Если возникло исключение IOException, выводится сообщение "Ошибка при чтении или записи файла.".**

**Класс CustomAgeException является пользовательским исключением, наследующимся от класса Exception. Он принимает два параметра: age (возраст) и path (путь к файлу), и переопределяет конструктор. В конструкторе создается экземпляр BufferedWriter с использованием пути к файлу path для записи сообщений об ошибке.**

**В зависимости от значения параметра age, выводится соответствующее сообщение о неверном возрасте в консоль и записывается в файл. Если age меньше 0, выводится сообщение age < 0 и записывается в файл. Если age больше 120, выводится сообщение age > 0 и записывается в файл. После записи сообщения в файл, происходит добавление пустой строки и закрытие BufferedWriter.**