**Министерство науки и высшего образования Российской Федерации**  
  
Федеральное государственное автономное образовательное учреждение  
высшего образования «Уральский федеральный университет  
имени первого Президента России Б.Н. Ельцина»  
  
Институт радиоэлектроники и информационных технологий – РТФ  
Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 11**по дисциплине «Курсовое проектирование Java»

**Тема: «Наследование. Обработка исключительных ситуаций»**

|  |  |
| --- | --- |
| Студент группы РИВ-410938у | М.О. Покуль |
| Преподаватель: | Н.А. Архипов |

**Екатеринбург 2024**

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc190897547)

[2 Описание задачи 4](#_Toc190897548)

[3 Ход выполнения работы. 5](#_Toc190897549)

[3.1.1 Пример 1 5](#_Toc190897550)

[3.1.2 Пример 2 6](#_Toc190897551)

[3.1.3 Пример 3 7](#_Toc190897552)

[3.1.4 Пример 4 8](#_Toc190897553)

[3.1.5 Пример 5 9](#_Toc190897554)

[3.1.6 Пример 6 10](#_Toc190897555)

[3.1.7 Пример 7 11](#_Toc190897556)

[3.1.8 Пример 8 12](#_Toc190897557)

[3.1.9 Пример 9 13](#_Toc190897558)

[3.1.10 Пример 10 14](#_Toc190897559)

[3.1.11 Пример 11 15](#_Toc190897560)

[3.1.12 Пример 12 16](#_Toc190897561)

[3.1.13 Пример 13 17](#_Toc190897562)

[3.1.14 Пример 14 18](#_Toc190897563)

[3.2 Задания из таблицы 2. 19](#_Toc190897564)

[3.2.1 Задание 1 19](#_Toc190897565)

[3.2.2 Задание 2 21](#_Toc190897566)

[3.2.3 Задание 3 22](#_Toc190897567)

[4 Вывод 24](#_Toc190897568)

1. Постановка задачи

Знакомство с иерархией классов исключений и получение навыков обработки ошибок.

1. Описание задачи
2. Воспроизвести примеры 1-14 лабораторной работы и при отправке в удаленный репозиторий исправить ошибки.
3. Выполнить все задания из таблицы 2:

* определить экспериментально, ошибки каких классов будут сгенерированы;
* создать обработчики исключительных ситуаций с использованием выявленных классов и всех секций конструкции обработчика с соответствующими сообщениями, позволяющими корректно выполнить программу.

1. Ход выполнения работы.
   1. **Примеры 1-14 лабораторной работы и при отправке в удаленный репозиторий исправить ошибки.**
      1. Пример 1

package Lab13.Task1;  
  
public class example1 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("Непроверяемая ошибка");  
 } catch (RuntimeException e) {  
 System.*out*.println("1 " + e);  
 }  
 System.*out*.println("2");  
 }  
}

* + 1. Пример 2

package Lab13.Task1;  
public class example2 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("Непроверяемая ошибка");  
 } catch (Exception e) {  
 System.*out*.println("1 " + e);  
 }  
 System.*out*.println("2");  
 }  
}

* + 1. Пример 3

package Lab13.Task1;  
  
public class example3 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 } catch (RuntimeException e) {  
 System.*out*.println("2");  
 } catch (Exception e) {  
 System.*out*.println("3");  
 }  
 System.*out*.println("4");  
 }  
}

* + 1. Пример 4

package Lab13.Task1;  
  
public class example4 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 } catch (Exception e) {  
 System.*out*.println("2");  
 } catch (Error e) {  
 System.*out*.println("3");  
 }  
 System.*out*.println("4");  
 }  
}

* + 1. Пример 5

package Lab13.Task1;  
  
public class example5 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 }  
 System.*out*.println("2");  
 }  
}

* + 1. Пример 6

package Lab13.Task1;  
  
public class example6 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new NullPointerException("ошибка");  
 } catch (ArithmeticException e) {  
 System.*out*.println("1");  
 } catch (Exception e) {  
 System.*out*.println("2");  
 } catch (RuntimeException e) {  
 System.*out*.println("3");  
 }  
 System.*out*.println("4");  
 }  
}

* + 1. Пример 7

package Lab13.Task1;  
  
public class example7 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new NullPointerException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 throw new ArithmeticException();  
 } catch (ArithmeticException e) {  
 System.*out*.println("2");  
 }  
 System.*out*.println("3");  
 }  
}

* + 1. Пример 8

package Lab13.Task1;  
  
public class example8 {  
 public static int m() {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException();  
 } finally {  
 System.*out*.println("1");  
 }  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println(*m*());  
 }  
}

* + 1. Пример 9

package Lab13.Task1;  
  
public class example9 {  
 public static int m() {  
 try {  
 System.*out*.println("0");  
 return 55;  
 } finally {  
 System.*out*.println("1");  
 }  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println(*m*());  
 }  
}

* + 1. Пример 10

package Lab13.Task1;  
  
public class example10 {  
 public static int m() {  
 try {  
 System.*out*.println("0");  
 return 15;  
 } finally {  
 System.*out*.println("1");  
 return 20;  
 }  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println(*m*());  
 }  
}

* + 1. Пример 11

package Lab13.Task1;  
  
public class example11 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new NullPointerException("ошибка");  
 } catch (NullPointerException e) {  
 System.*out*.println("1");  
 } finally {  
 System.*out*.println("2");  
 }  
 System.*out*.println("3");  
 }  
}

* + 1. Пример 12

package Lab13.Task1;  
  
public class example12 {  
 public static void m(String str, double chislo) {  
 if (str == null) {  
 throw new IllegalArgumentException("Строка введена неверно");  
 }  
 if (chislo > 0.001) {  
 throw new IllegalArgumentException("Неверное число");  
 }  
 }  
  
 public static void main(String[] args) {  
 *m*(null, 0.000001);  
 }  
}

* + 1. Пример 13

package Lab13.Task1;  
  
public class example13 {  
 public static void main(String[] args) {  
 try {  
 int l = args.length;  
 System.*out*.println("Размер массива= " + l);  
  
 int h = 10 / l;  
 args[l +1] = "10";  
 } catch (ArithmeticException e) {  
 System.*out*.println("Деление на ноль");  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println("Индекс не существует");  
 }  
 }  
}

* + 1. Пример 14

package Lab13.Task1;  
  
public class example14 {  
 public static void m(int x) throws ArithmeticException {  
 int h = 10 / x;  
 }  
  
 public static void main(String[] args) {  
 try {  
 int l = args.length;  
 System.*out*.println("Размер массива= " + l);  
 *m*(l);  
 } catch (ArithmeticException e) {  
 System.*out*.println("Ошибка: Деление на ноль");  
 }  
 }  
}

* 1. Задания из таблицы 2.
     1. Задание 1

package Lab13.Task2;  
  
import java.util.Arrays;  
import java.util.InputMismatchException;  
import java.util.NoSuchElementException;  
import java.util.Scanner;  
  
public class Task1 {  
 public static void main(String[] args) {  
 try {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите размер массива");  
 int size = in.nextInt();  
 System.*out*.println();  
  
 int[] arr = new int[size];  
  
 for (int i = 0; i < size; i++) {  
 System.*out*.println("Введите "+i+" элемент:");  
 arr[i] = in.nextInt();  
 }  
  
 System.*out*.println("\nМассив arr : ");  
 System.*out*.println(Arrays.*toString*(arr));  
   
 System.*out*.println("\nСреднее : ");  
 System.*out*.println(*avgNum*(arr));  
 } catch (InputMismatchException e) {  
 System.*out*.println("Введено не целое число!");  
 } catch (NoSuchElementException e) {  
 System.*out*.println("Нет положительных элементов!");  
 }  
 }  
  
 public static double avgNum(int[] arr){  
 arr = Arrays.*stream*(arr).filter(x -> x >= 0).toArray();  
 double res = Arrays.*stream*(arr).average().getAsDouble();  
 return res;  
 }  
}

* + 1. Задание 2

package Lab13.Task2;  
  
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
public class Task2 {  
 public static void main(String[] args) {  
 try {  
 Scanner in = new Scanner(System.*in*);  
 int[][] arr = new int[10][10];  
  
 for (int i = 0; i < 10; i++) {  
 for (int j = 0; j < 10; j++) {  
 arr[i][j] = i + j;  
 }  
 }  
  
 System.*out*.println("Введите столбец");  
 int col = in.nextInt();  
 System.*out*.println();  
   
 for (int i = 0; i < 10; i++) {  
 System.*out*.println(arr[i][col - 1]);  
 }  
  
 } catch (InputMismatchException e) {  
 System.*out*.println("Введено не целое число!");  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println("Такого стоблца не существует!");  
 }  
 }  
}

* + 1. Задание 3

package Lab13.Task2;  
  
import java.util.Arrays;  
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
public class Task3 {  
 public static void main(String[] args) {  
 try {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите размер массива");  
 int size = in.nextInt();  
 System.*out*.println();  
  
 byte[] arr = new byte[size];  
  
 for (int i = 0; i < size; i++) {  
 System.*out*.println("Введите "+i+" элемент:");  
 arr[i] = in.nextByte();  
 }  
  
 System.*out*.println("\nМассив arr : ");  
 System.*out*.println(Arrays.*toString*(arr));  
   
 System.*out*.println("\nСумма : ");  
 System.*out*.println(*sum*(arr));  
 } catch (InputMismatchException e) {  
 System.*out*.println("Введённые данные не являются byte!");  
 } catch (ArithmeticException e) {  
 System.*out*.println("Сумма выходит за границы!");  
 }  
 }  
  
 public static byte sum(byte[] arr){  
 int sum = 0;  
 for (byte b : arr) {  
 sum += b;  
 }  
  
 if (sum < Byte.*MIN\_VALUE* || sum > Byte.*MAX\_VALUE*) {  
 throw new ArithmeticException();  
 }  
  
 return (byte) sum;  
 }  
}

1. Вывод

Познакомились с иерархией классов исключений и получение навыков обработки ошибок.