ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования
«Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

Кафедра Безопасности информационных систем

Отчет по лабораторной работе №6

«Разработка программы корректно отрабатывающей исключения.»

По дисциплине: «Кроссплатформенное программирование»

Цель работы:

1. Изучение общих принципов программирования в среде java.
2. Написание приложения, обрабатывающего исключения.

Теоретическая часть:

- В мире программирования возникновение ошибок и непредвиденных ситуаций при выполнении программы называют **исключением**. В программе исключения могут возникать в результате неправильных действий пользователя, отсутствии необходимого ресурса на диске, или потери соединения с сервером по сети. Причинами исключений при выполнении программы также могут быть ошибки программирования или неправильное использование API. В отличие от нашего мира, программа должна четко знать, как поступать в такой ситуации. Для этого в Java предусмотрен механизм исключений.

- Обработка исключений в Java основана на использовании в программе следующих ключевых слов:

  try – определяет блок кода, в котором может произойти исключение;

  catch – определяет блок кода, в котором происходит обработка исключения;

  finally – определяет блок кода, который является необязательным, но при его наличии выполняется в любом случае независимо от результатов выполнения блока try.

Эти ключевые слова используются для создания в программном коде специальных обрабатывающих конструкций: try{}catch, try{}catch{}finally, try{}finally{}.

throw – используется для возбуждения исключения;

throws – используется в сигнатуре методов для предупреждения, о том что метод может выбросить исключение.

- Создание исключения

При исполнении программы исключение генерируется JVM или вручную, с помощью оператора throw. При этом в памяти создается объект исключения и выполнение основного кода программы прерывается, а обработчик исключений **JVM** пытается найти способ обработать исключение.

- Обработка исключения

Создание блоков кода, для которых мы предусматриваем обработку исключений в Java, производится в программе с помощью конструкций try{}catch, try{}catch{}finally, try{}finally{}.
При возбуждении исключения в блоке try обработчик исключения ищется в следующем за ним блоке catch. Если в catch есть обработчик данного типа исключения – управление переходит к нему. Если нет, то JVM ищет обработчик этого типа исключения в цепочке вызовов методов до тех пор, пока не будет найден подходящий catch.

Ход работы:

```
package Isklucheniya;

import java.io.*;
import java.util.*;

public class Isklucheniya
{

  static Scanner in = new Scanner(System.in);

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

     int w=0;

     while (w!=6)
       {
         System.out.println();
         System.out.println("-----------------------------------------------");
         System.out.println("1 - Генерируем пример ошибки ввода/вывода");
         System.out.println("2 - Переполняем масив");
         System.out.println("3 - Откроем отстутствующий *.txt файл");
         System.out.println("4 - Поделим число на 0");
         System.out.println("5 - Вычислим факториал любого числа");
         System.out.println("6 - Выход");
         System.out.println("Ваш выбор =  ");
         int q = in.nextInt();
         System.out.println();
         switch (q)
        {

        case 1:
          {
          INPUT_AND_OUTPUT_ERROR();
          }
          break;


        case 2:
          {
          PEREPOLNENIE_MASSIVA();
          }
          break;


        case 3:
```

```
        {
        OTKRYTIE_OTSUT_FILE();
        }
        break;


      case 4:
        {
        DELENIE_NA_0();
        }
        break;

      case 5:
        {
        FACTORIAL();
        }
        break;

      case 6:
        {
        w=6;
        System.out.println("EXIT");
        }
        break;

      default:
         {
         w=0;
         System.out.println("Not found");
         }
         break;

      }

      }
  }



static void INPUT_AND_OUTPUT_ERROR()
{
       try
       {
          FileInputStream fin = null;
          int i1;

          try
          {
           fin = new FileInputStream("C://XXX//XXX.txt");
           fin.close();
           while((i1=fin.read())!=-1);
           {
           System.out.println((char)i1);
           }
          }
          catch(IOException e1)
          {
           System.out.println("Исключение: "+ e1);
          }


          FileOutputStream fin1 = null;
          String Str="YYY";
          int i2;
```

```
            try
            {
             fin1 = new FileOutputStream("C://XXX//XXX.txt");
             fin1.close();
             fin1.write(Str.getBytes());
            }
            catch(IOException e1)
            {
             System.out.println("Исключение: "+ e1);
            }
          }
          finally
          {
           System.out.println("Complete");
          }
  }



static void PEREPOLNENIE_MASSIVA()
{
  try
            {
            int[] array = new int[4];
            array[0]=4; array[1]=8; array[2]=10; array[3]=11;
            System.out.println("Существует массив array[], состоящий из 4
элементов: " + array[0] +", "+array[1] +", "+array[2] +", "+array[3]);
            System.out.println("Пробуем добавить 5 элемент, array[5] = ");
            int m=in.nextInt();
            array[5]=m;
            }
            catch(ArrayIndexOutOfBoundsException e2)
            {
            System.out.println("Исключение:" + e2);
            }
             finally
            {
            System.out.println("Complete");
            }
}



static void OTKRYTIE_OTSUT_FILE()
{
 System.out.println("Корректная директория: " );
            String source=in.next();
            System.out.println("Имя несуществующего открываемого файла,
который будет открываться неизвестным редактором");
            String fileopen =in.next();
            try
             {
             Runtime runtime = Runtime.getRuntime();
             Process process = runtime.exec("C\\ " +source + "/" +fileopen
+".txt");
             }
            catch(IOException e3)
             {

             System.out.println("Исключение:"+e3);
             }
            finally
             {
             System.out.println("Complete");
```

```
            }
}


static void DELENIE_NA_0()
{
  System.out.println("Число, которое делим на 0: " );
            int m =in.nextInt();
            try
             {
             double n=m/0;
             }
            catch(ArithmeticException e4)
             {

             System.out.println("Исключение:"+e4);
             }
            finally
             {
             System.out.println("Complete");
             }
}



static void FACTORIAL()
{
   System.out.println("Введите число: ");
   int m=in.nextInt();
         try
           {
           int result = getFactorial(m);
           System.out.println(result);
           }
           catch(FactorialException ex){
           System.out.println(ex.getMessage());
           }
           finally
           {
           System.out.println("Complete");
           }
}




public static int getFactorial(int num) throws FactorialException
    {
      int result=1;
        if(num<0) throw new FactorialException("The number is less than 0");

        for(int i=1; i<=num;i++)
         {
         result*=i;
         }
        return result;
    }



static class FactorialException extends Exception
{
    public FactorialException(String message)
    {
```

```
        super(message);
    }
}

}
```

```
тся найти указанный путь)
тся найти указанный путь)
```

```
ным редактором

or=2, Не удается найти указанный файл
```

Выводы:

- Изучены общие принципы программирования в среде java.
- Написано приложение, обрабатывающее исключения.