*Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана  (национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

ДИСЦИПЛИНА Языки программирования для работы с большими данными

**Отчет**

**по лабораторной работе № 5**

**Вариант 5**

Студент гр. ИУ6-21М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_**Войтов Д.Ю.\_\_\_

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2020

**ЗАДАНИЕ 1**

*Условие*

Выполнить задания на основе варианта 1 лабораторной работы 4, контролируя состояние потоков ввода/вывода. При возникновении ошибок, связанных с корректностью выполнения математических операций, генерировать и обрабатывать исключительные ситуации. Предусмотреть обработку исключений, возникающих при нехватке памяти, отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д

Задание: Создать класс Календарь с внутренним классом, с помощью объектов которого можно хранить информацию о выходных и праздничных днях.

*Решение*

Класс Календарь с внутренним классом приведен ниже:

|  |
| --- |
| package com.company;  import java.util.ArrayList; import java.util.InputMismatchException; import java.util.List;  public class Calendar {   private List<Holiday> holidays = new ArrayList<Holiday>();  private char[] map = new char[366]; // w - раб день, h - вых   public Calendar() {  for (int i = 0; i < 366; i++) {  if (i % 7 == 0) //Вск = выходной. Правда здесь календарь на один и тот же год, где 1 января = вск.  {  map[i] = 'h';   } else {  map[i] = 'w';  }  }  }   public class Holiday {   private String name;  private int date\_strt;  private int date\_end;   public Holiday(String s, int d1, int d2) {   this.name = s;  this.date\_end = d2;  this.date\_strt = d1;   }   }   public void AddHoliday(String s, int d1, int d2) {  try {  Holiday h = new Holiday(s, d1, d2);  holidays.add(h);  for (int i = d1; i < d2; i++) {   map[i] = 'h';   }  }catch (InputMismatchException e1) {  System.out.println("You make some mistakes, bro :C");  }   }   public void Show() {   System.out.println(map);   }   public void GetTypeDay(int i) { //Выводит тип дня (выходной или рабочий)  try {  System.out.println(map[i]);  } catch (IndexOutOfBoundsException e1) {  System.out.println("Wrong day, bro :C");  }  }  } |

Основная программа приведена ниже:

|  |
| --- |
| package com.company;  import java.util.InputMismatchException; import java.util.Scanner;  public class Main {   public static void main(String[] args) {   Scanner scan = new Scanner(System.in);  Calendar cal = new Calendar();  cal.Show();  System.out.println("Add new Holidays:");  System.out.println("HolidayName:");  String s = scan.nextLine();  int d1 = 0, d2 = 0;  try {  System.out.println("Start:");  d1 = scan.nextInt();  System.out.println("End:");  d2 = scan.nextInt();  } catch (InputMismatchException e1) {  System.out.println("You make some mistakes, bro :C");  }   try {  cal.AddHoliday(s, d1, d2);  } catch (IndexOutOfBoundsException e1) {  System.out.println("You make some mistakes, bro :C");  }  cal.Show();  int k = 0;   System.out.println("Get type of day:");  try {  k = scan.nextInt();  } catch (InputMismatchException e1) {  System.out.println("You make some mistakes, bro :C");  }  cal.GetTypeDay(k);   } } |

**ЗАДАНИЕ 2**

*Условие*

Выполнить задания из варианта 2 лабораторной работы 4, реализуя собственные обработчики исключений и исключения ввода/вывода.

Задание: Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов: interface Mobile <- abstract class Siemens Mobile <- class Model.

*Решение*

Основной метод, содержащий обработку исключений, приведен ниже:

|  |
| --- |
| package com.company;  import java.util.Scanner;  public class Main {   public static void main(String[] args) throws Exept {    Scanner scan = new Scanner(System.in);  char c1, c2;  c1='0';  c2='1';  System.out.println("Input ID");  String s = scan.nextLine();  System.out.println("Input chips");  String cs = scan.nextLine();   try { //Обработка стандартного исключения   c1 = cs.charAt(0);  c2 = cs.charAt(1);  ModelT me\_phone = new ModelT(s, c1, c2);  me\_phone.Info();  me\_phone.StartScreen();  } catch (IndexOutOfBoundsException e1) {  System.out.println("Wrong input!");  }  } } |

**ЗАДАНИЕ 3**

*Условие*

В следующих заданиях требуется ввести последовательность строк из текстового потока и выполнить указанные действия. Найти в строке наибольшее число цифр, идущих подряд.

*Решение*

Решение приведено на листинге ниже:

|  |
| --- |
| package com.company;  import java.io.File; import java.io.FileNotFoundException; import java.util.Scanner;  public class Main {   public static void main(String[] args) throws FileNotFoundException {   File input = new File("input.txt");   try (Scanner scan = new Scanner(input)) {  // Ищем наибольшее число цифр идущих подряд  int max = 0;  while (scan.hasNextLine()) {  String s = scan.nextLine();  System.out.println(s);  int tmp = 0;  for (int i = 0; i < s.length(); i++) {  if (Character.isDigit(s.charAt(i))) {  // Если цифра, наращиваем счетчик  tmp++;  }  else {  // Если нет - проверяем, больше ли результат, и сбрасываем  if (tmp > max) max = tmp;  tmp = 0;  }  }  if (tmp > max) max = tmp;  tmp=0;  }  System.out.println("Most digits: " + max);   }  catch (FileNotFoundException exp) {  System.out.println("I don't see that file, bro :C");  }  } } |

**ЗАДАНИЕ 4**

*Условие*

При выполнении следующих заданий для вывода результатов создавать новую директорию и файл средствами класса File. Файл содержит символы, слова, целые числа и числа с плавающей запятой. Определить все данные, тип которых вводится из командной строки.

*Решение*

Приведем решение на листинге ниже:

|  |
| --- |
| package com.company;  import java.io.File; import java.io.FileNotFoundException; import java.io.FileWriter; import java.io.IOException; import java.util.Scanner;  public class Main {   public static void main(String[] args) {   File input = new File("input.txt");  Scanner scan\_ = new Scanner(System.in);  System.out.println("Enter the desired type:");  System.out.println("Word: 'dasldasd', 'ab', etc.");  System.out.println("Int: '1','24','124', etc.");  System.out.println("Float: '1.2', '0.0', '132.3123', etc.");  System.out.println("Char: 'a', 'b', '$', etc.");  String type = scan\_.nextLine();   File dir = new File("outputs");  Boolean dir\_cr = dir.mkdirs();  String path = "outputs\\output\_" + type + ".txt";  File output = new File(path);   try {   FileWriter myWriter = new FileWriter(path);   try (Scanner scan = new Scanner(input)) {   while (scan.hasNextLine()) {  String s = scan.nextLine();   // Определяем тип  String type\_of\_line = "Int";  for (int i = 0; i < s.length(); i++) {  if (s.charAt(i) == '.' && type\_of\_line.equals("Int")) {  type\_of\_line = "Float";  }  if (!Character.isDigit(s.charAt(i)) && s.charAt(i) != '.') {  type\_of\_line = "Word";  }  }  if (type\_of\_line.equals("Word") && s.length() == 1) {  type\_of\_line = "Char";  }  // Тип определен   if (type\_of\_line.equals(type)) {  // Выводим подобранные строки в файл  myWriter.write(s+'\r'+'\n');  }  }   }  catch (FileNotFoundException exp) {   System.out.println("I don't see input file, bro :C");   }  myWriter.close();   } catch (IOException e) {   System.out.println("Something with output, bro :C");   }  } } |

**ВЫВОД**

В ходе выполнения данной работы было получено представление об обработке исключений, также были использованы операции работы с файлами: считывание файла, запись в файл, обработка исключений файла, создание новой директории.