《Java语言程序设计》课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机科学与技术 | 年级 | 2017 | 班级 | 计工本1704 |
| 学生姓名 | 付泽坤 | 指导老师 | 李焱 | 时间 | 2019.05.30 |

|  |  |
| --- | --- |
| 实验名称 | 异常与IO |
| 实  验  目  的  及  要  求 | 目的：  了解熟悉顺序Java程序设计的形式，编写完整Java程序。  要求：   * 掌握异常类。 * 掌握异常捕获机制。 * 掌握捕获异常。 * 会用Java编写完整的程序。 |
| 实  验  环  境 | Microsoft Windows 7 with SP1专业版（简体中文）32位  JDK 1.8  Eclipse 2017  或者Jcreator |
| 实  验  内  容 | 请按照要求编写出完整程序   * 第12章编程练习题: 12.3、12.5、12.8、12.13-14、12.32 |
| 实  验  步  骤  或  实  验  方  案 | **package** stu.ltr.homework;  **import** java.util.Scanner;  **import** java.util.Random;  **public** **class** zuoye {    **public** **static** **void** main(String [] args)  {  Scanner input = **new** Scanner(System.***in***);  **int**[] **numbers** = *creatNumberArray*();  **while** (**true**)  {  **try**  {  System.***out***.print("Enter a index to find the number: ");  **int** index = input.nextInt();  System.***out***.println(numbers[index]);  }  **catch** (ArrayIndexOutOfBoundsException ex)  {  System.***out***.println("Out of Bounds");  }  input.nextLine();  }  }  **public** **static** **int**[] creatNumberArray()  {  Random num = **new** Random(9);  **int**[] numbers = **new** **int**[100];  **for** (**int** i = 0; i < 100; i++)  {  numbers[i] = num.nextInt(100) + 1;  }  **return** numbers;  }  }  12.5  （1）  **public** **class** IllegalTriangleException **extends** Exception{  **private** **double** side1;  **private** **double** side2;  **private** **double** side3;  **public** IllegalTriangleException(**double** side1, **double** side2, **double** side3)  {  **super**("Invaoide sides " + side1 + " " + side2 + " " + side3);  **this**.side1 = side1;  **this**.side2 = side2;  **this**.side3 = side3;  }  **public** **double** getSide1()  {  **return** side1;  }  **public** **double** getSide2()  {  **return** side2;  }  **public** **double** getSide3()  {  **return** side3;  }  }  （2）  **public** **class** TriangleUseIllegalTrianglException {  **private** **double** side1;  **private** **double** side2;  **private** **double** side3;  **public** TriangleUseIllegalTrianglException()  **throws** IllegalTriangleException  {  **this**(1.0, 1.0, 1.0);  }  **public** TriangleUseIllegalTrianglException(**double** side1, **double** side2, **double** side3)  **throws** IllegalTriangleException  {  setSides(side1, side2, side3);  }  **public** **void** setSides(**double** side1, **double** side2, **double** side3)  **throws** IllegalTriangleException  {  **if** ((side1 + side2 <= side3) || (side1 - side2 >= side3))  {  **throw** **new** IllegalTriangleException(side1, side2, side3);  }  **else**  {  **this**.side1 = side1;  **this**.side2 = side2;  **this**.side3 = side3;  }  }  **public** **void** setSide1(**double** side1) **throws** IllegalTriangleException  {  setSides(side1, **this**.side2, **this**.side3);  }  **public** **void** setSide2(**double** side2) **throws** IllegalTriangleException  {  setSides(**this**.side1, side2, **this**.side3);  }  **public** **void** setSide3(**double** side3) **throws** IllegalTriangleException  {  setSides(**this**.side1, **this**.side2, side3);  }  **public** **double** getSide1()  {  **return** side1;  }  **public** **double** getSide2()  {  **return** side2;  }  **public** **double** getSide3()  {  **return** side3;  }  **public** **double** getArea()  {  **double** s = (side1 + side2 + side3) / 2;  **return** Math.*sqrt*(s \* (s - side1) \* (s - side2) \* (s - side3));  }  **public** **double** getPerimeter()  {  **return** (side1 + side2 + side3);  }  }  (3)  **import** java.util.Scanner;  **public** **class** TestTriangleUseIllegalTrianglException {  **public** **static** **void** main(String [] args)  {  Scanner input = **new** Scanner(System.***in***);  **while** (**true**)  {  System.***out***.print("Enter three sides for a triangle: ");  **try**  {  **double** side1 = input.nextDouble();  **double** side2 = input.nextDouble();  **double** side3 = input.nextDouble();  TriangleUseIllegalTrianglException triangle =  **new** TriangleUseIllegalTrianglException(side1, side2, side3);  System.***out***.println("The area is " + triangle.getArea());  System.***out***.println("The perimeter is " + triangle.getPerimeter());  }  **catch** (IllegalTriangleException ex)  {  System.***out***.println(ex.getMessage());  System.***out***.println("Try again.");  }  input.nextLine();  }  }  }  12.8  (1)  **public** **class** HexFormatException **extends** Exception  {  String hexString;  **public** HexFormatException(String hexString)  {  **super**("HexFormatException: " + hexString);  **this**.hexString = hexString;  }  **public** String getHexString()  {  **return** hexString;  }    }  (2)  **import** java.util.Scanner;  **public** **class** HexToDecUseHexFormatException  {  String hexString;  **public** HexToDecUseHexFormatException(String hexString)  **throws** HexFormatException  {  setHexString(hexString);  }  **public** **void** setHexString(String hexString)  **throws** HexFormatException  {  **for** (**int** i = 0; i < hexString.length(); i++)  {  **if** (judge(hexString.charAt(i)) == -1)  {  **throw** **new** HexFormatException(hexString + " is not a hex string");  }  }  **this**.hexString = hexString;  }  **public** String getHexString()  {  **return** hexString;  }  **private** **int** judge(**char** ch)  {  **if** ((ch >= '0') && (ch <= '9'))  {  **return** (ch - '0');  }  **else** **if** ((ch >= 'A') && (ch <= 'F'))  {  **return** (ch - 'A' + 10);  }  **else**  **return** -1;  }  **public** **int** hexToDec()  {  **int** result = 0;  **for** (**int** i = hexString.length() - 1; i >= 0; i--)  {  result += judge(hexString.charAt(i)) \* (Math.*pow*(16, (hexString.length() - i - 1)));  }  **return** result;  }  }  (3)  **import** java.util.Scanner;  **public** **class** TestHexFormatException {  **public** **static** **void** main(String [] args)  {  Scanner input = **new** Scanner(System.***in***);  **try**  {  System.***out***.print("Enter a hex string: ");  HexToDecUseHexFormatException hex =  **new** HexToDecUseHexFormatException(input.next());  System.***out***.println("The decimal value for hex number "  + hex.getHexString() + " is " + hex.hexToDec());  }  **catch** (HexFormatException ex)  {  System.***out***.println(ex.getMessage());  }  }  }  12.13  **import** java.util.Scanner;  **import** java.io.File;  **public** **class** Statistics {  **public** **static** **void** main(String [] args) **throws** Exception  {  **if** (args.length != 1)  {  System.***out***.println("Useage: file name");  }  **int** countWords = 0;  **int** countLetters = 0;  **int** countLines = 0;  Scanner input = **new** Scanner(**new** File(args[0]));  **while** (input.hasNextLine())  {  String strLine = input.nextLine();    **for** (**int** i = 0; i < strLine.length(); i++)  {  **if** (Character.*isLetter*(strLine.charAt(i)))  {  countLetters++;  }  }    **for** (**int** i = 0; i < strLine.length() - 1; i++)  {  **if** (Character.*isLetter*(strLine.charAt(i)) && strLine.charAt(i + 1) == ' ')  {  countWords++;  }  }  **if** (Character.*isLetter*(strLine.charAt(strLine.length() - 1)))  {  countWords++;  }  countLines++;  }  System.***out***.println("File " + args[0]);  System.***out***.println(countLetters + " character");  System.***out***.println(countWords + " words");  System.***out***.println(countLines + " lines");  input.close();  }  }  12.14  **import** java.util.Scanner;  **import** java.io.File;  **public** **class** TheScoreInFile {  **public** **static** **void** main(String [] args) **throws** Exception  {  Scanner input = **new** Scanner(System.***in***);  System.***out***.print("Enter the name of a file: ");  String fileName = input.nextLine();  File file = **new** File(fileName);  **if** (!file.exists())  {  System.***out***.println(fileName + " is not exists");  System.*exit*(1);  }  **double** totalScore = 0.0;  **int** count = 0;  Scanner in = **new** Scanner(file);  System.***out***.println("loading...");  **while** (in.hasNextInt())  {  System.***out***.println("loading...");  totalScore += in.nextDouble();  in.skip(" ");  count++;  }  input.close();  in.close();  System.***out***.println("Total scores is " + totalScore);  System.***out***.println("Average is " + (totalScore / count));  }  } |
| 调  试  过  程  及  实  验  结  果 |  |
| 总  结 |  |
| 附  录 |  |