|  |
| --- |
| **《Java面向对象程序设计SSD3》**  **实验报告**  项目名称 实验1《基本程序设计、选择、循环》  专业班级 软件工程2005班  学 号 8209200504  姓 名 李均浩  **实验成绩：**  **批阅教师：邝砾**  2021年 4 月 20 日 |

**中南大学计算机学院实验报告**

**课程名称** Java面向对象程序设计SSD3

**实验项目名称** 实验1《基本程序设计、选择、循环》

**学生姓名** 李均浩 **专业班级** 软件工程2005班 **学号** 8209200504

**实验成绩**  **日期** 2021年4月21日

实验学时： 2

每组人数： 1

实验类型： 1 （1：基础性 2：综合性 3：设计性 4：研究性）

实验要求： 1 （1：必修 2：选修 3：其它）

实验类别： 2 （1：基础 2：专业基础 3：专业 4：其它）

**一、实验目的**

学习Java基本程序编写的方法，熟悉选择、循环等控制结构。

**二、实验内容**

1. （P60, 2.6）编写程序，读取一个在0和1000之间的整数，并将该整数的各位数字相加。例如：整数932，各位数字之和为14。如果输入的整数不在0-1000范围内，告知用户输入有误。

提示：利用%和/分解数字。

2. （P62, 2.19）编写程序，提示用户输入三角形的三个点（x1, y1）,(x2, y2), (x3, y3)，然后显示它的面积。计算三角形面积的公式是：

s=(s1+s2+s3)/2;

面积 = 



其中s1,s2, s3分别为三角形三边的长度。

输入示例：

Enter three points for a triangle: 1.5 -3.4 4.6 5 9.5 -3.4

The area of the triangle is 33.6

提示：使用Math类中的方法进行相关运算，见102页4.2.2节。

3. （P96，3.24）编写程序，模拟从一副52张的牌中选择一张牌。程序应显示牌的大小（Ace、2、3、4、5、6、7、8、9、10、Jack、Queen、King）以及牌的花色（Clubs、Diamond、Heart、Spades）。

提示: 设1-13为黑桃Ace至King，14-26为红桃Ace至King，27-39为梅花Ace至King，40-52为方片Ace至King。随机生成1-52内的一个随机整数，根据以上规则判断并提示牌的花色和数字。

4.（P164，5.11）编写程序，显示从100到200之间所有能被5或6整除，但不能被两者同时整除的数，每行显示10个数。（提示：建议使用模块化代码的方式）

三、实验要求：

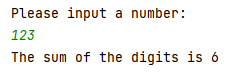
要求每个学生独立完成实验任务。

四、实验报告

**1. 实验结果与分析**

**第一题：**

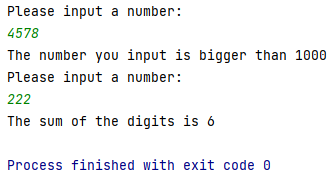
测试点( a )：123

 ( 1 )测试结果：

( 2 )结果分析

输入数字不大于1000，每个数位数字相加为1+2+3=6，输出结果符合预期。

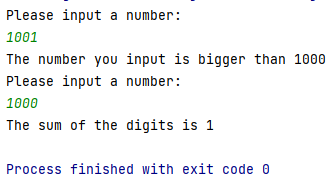
测试点( b )：4578 222

 ( 1 )测试结果：

( 2 )结果分析：

第一个输入的数字大于1000，输出错误提示，重新输入了一个小于1000且大于0的数字：222，输出2+2+2=6，符合预期。

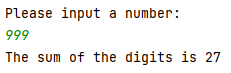
测试点( c )：1001 1000

 ( 1 )测试结果：

( 2 )结果分析：

第一个输入的数字大于1000，输出错误提示，重新输入了1000输出1+0+0+0=1，符合预期。

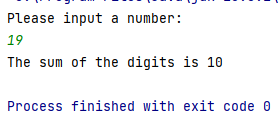
测试点( d )：999

 ( 1 )测试结果：

( 2 )结果分析：

输入的数字小于1000且大于0，输出数位的和9+9+9=27，符合预期。

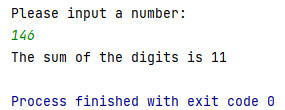
测试点( e )：19

 ( 1 )测试结果：

( 2 )结果分析：

输入的数字小于1000且大于0，输出数位的和1+9=10，符合预期。

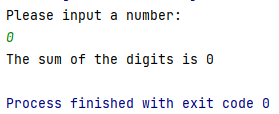
测试点( e )：146

 ( 1 )测试结果：

( 2 )结果分析：

输入的数字小于1000且大于0，输出数位的和1+4+6=11，符合预期。

测试点( e )：0

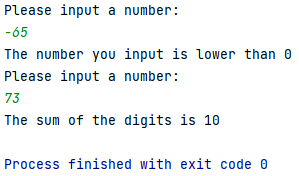
 ( 1 )测试结果：

( 2 )结果分析

输入的数字小于1000且大于0，输出数位的和0=0，符合预期。

测试点( f )：-65 73

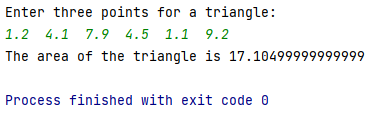
( 1 )测试结果：

 ( 2 )结果分析：

第一个输入的数字小于0，输出错误提示，重新输入，第二次输入的数字为73小于1000且大于0，数位相加为7+3=10，符合预期。

**第二题：**

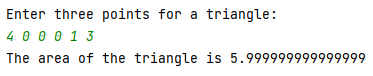
测试点( a )：1.2 4.1 7.9 4.5 1.1 9.2

 ( 1 )测试结果：

( 2 )结果分析：

利用坐标点计算出三边长度，利用三边长度公式求出面积，符合预期。

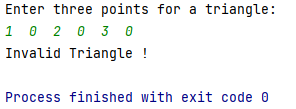
测试点( b )：4 0 0 0 1 3

 ( 1 )测试结果：

( 2 )结果分析：

利用坐标点计算出三边长度，利用三边长度公式求出面积，符合预期。

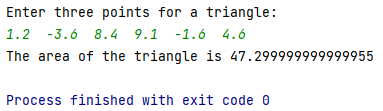
测试点( c )：1 0 2 0 3 0

 ( 1 )测试结果：

( 2 )结果分析：

输入的三个顶点不符合三角形的定义，位于一条直线上，输出提示三角形非法，符合预期。

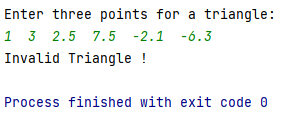
测试点( d )：1.2 -3.6 8.4 9.1 -1.6 4.6

 ( 1 )测试结果：

( 2 )结果分析：

利用坐标点计算出三边长度，利用三边长度公式求出面积，符合预期。

测试点( e )：1 3 2.5 7.5 -2.1 -6.3

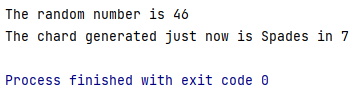
 ( 1 )测试结果：

( 2 )结果分析：

输入的三个顶点不符合三角形的定义，位于一条直线上，输出提示三角形非法，符合预期。

**第三题：**

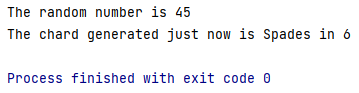
测试 ( a )：

 ( 1 )测试结果：

( 2 )结果分析：

产生了随机数46，属于40≤x≤52，花色应为Spades，是Spades的第7张牌，输出符合预期。

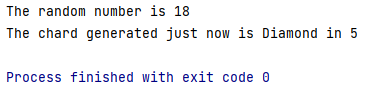
测试 ( b )：

 ( 1 )测试结果：

( 2 )结果分析：

产生了随机数45，属于40≤x≤52，花色应为Spades，是Spades的第6张牌，输出符合预期。

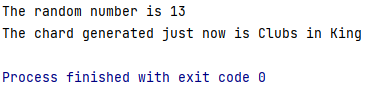
测试 ( c )：

 ( 1 )测试结果：

( 2 )结果分析：

产生了随机数18，属于14≤x≤26，花色应为Diamond，是Diamond的第5张牌，输出符合预期。

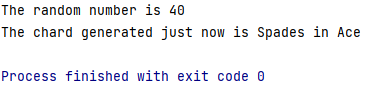
测试 ( d )：

 ( 1 )测试结果：

( 2 )结果分析：

产生了随机数13，属于1≤x≤13，花色应为Clubs，是Clubs的第13张牌，即King，输出符合预期。

测试 ( e )：

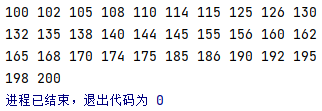
 ( 1 )测试结果：

( 2 )结果分析：

产生了随机数40，属于40≤x≤52，花色应为Spades，是Spades的第1张牌，即Ace，输出符合预期。

**第四题：**

测试：

 ( 1 )测试结果：

( 2 )结果分析：

从100开始进行遍历，到200（包括200）结束。遇到符合条件的数字立即输出，遍历过程中记录符合条件的数字总数，一旦总数对10取模的结果为0，进行换行，结果符合预期。

**2. 心得体会**

**第一题：**

本题输入的数据用字符串String接收，那么将每位数字进行相加就比较便捷，只需执行s.charAt(i) - '0'将数位的char类型转换为byte类型，再将每个数位的数值相加即可。确定输入的数是否大于0且不大于1000，使用了Integer.*parseInt*(s)这一方法，将字符串的所代表的数转为int类型进行判断。也使用实验指导书中所提示的方案进行编写，也非常便捷，可以直接使用输入的数进行是否符合大于0且不大于1000的判断。

**第二题：**

本题主要是利用三角形的三个顶点坐标和两点间距离公式计算出三边长，然后按照三角形的定义判定输入的三角形是否合法，若不合法输出提示，若合法则利用三边长和三角形面积公式计算并输出结果，其中开平方使用的是*Math*类中的*sqrt()*方法，以Math.*sqrt*(i)进行调用，平方计算使用Math类中的*pow()*方法，以Math.*pow*(a,b)进行调用。

**第三题：**

本题利用了Math.*random*()方法产生随机数，由于此方法产生的随机数范围是0≤x＜1，通过(int)(Math.*random*() \* 52) + 1 即可取到范围在1≤x≤52的随机整数。之后通过随机数落在的范围，确定模拟抽取扑克牌的花色，通过rand\_num % 13确定扑克牌的牌面大小，若为1，11，12，13，则相应的转换为Ace，Jack，Queen，King，最后进行字符串的合并，输出结果即可。其中使用了String.*valueOf()*方法将数字转换为了字符串类型。

**第四题：**

从100开始进行遍历，到200（包括200）结束。将当前的数字传入isValid(int i)方法，如果符合条件的立即输出，符合条件的数字总数自增，一旦总数对10取模的结果为0，即已经一行内已经输出了10个数字，进行换行。本题进行了模块化处理，将判断是否符合要求和输出功能分离，练习了方法调用，方法的按值传递。将程序进行模块化处理，方便对程序进行调试，让代码的可读性增强，也更方便于自顶向下、逐步求精的设计。

【**附源程序**】

import java.util.Scanner;  
  
public class Exp\_1 {  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
  
 *//接受输入的数字（串）* String s;  
 *//储存数位加和* byte sum = 0;  
  
 *//提示用户输入数字* System.*out*.println("Please input a number:");  
 s = input.nextLine();  
  
 *//如果大于1000，提示重新输入* while (Integer.*parseInt*(s) > 1000) {  
 System.*out*.println("The number you input is bigger than 1000");  
 System.*out*.println("Please input a number:");  
 s = input.nextLine();  
 }  
  
 *//如果小于0，提示重新输入* while (Integer.*parseInt*(s) < 0) {  
 System.*out*.println("The number you input is lower than 0");  
 System.*out*.println("Please input a number:");  
 s = input.nextLine();  
 }  
  
 *//求数位和* for (byte i = 0; i < s.length(); ++i) {  
 sum += (s.charAt(i) - '0');  
 }  
  
 *//输出结果* System.*out*.println("The sum of the digits is " + sum);  
 }  
}

源代码1.1 Exp\_1.java

源代码1.2 Exp\_2.java

import java.util.Scanner;  
  
public class Exp\_2 {  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
  
 *//储存输入的坐标* double x1, y1, x2, y2, x3, y3;  
 *//储存三边长，参数s* double s1, s2, s3, s;  
  
 *//输入* System.*out*.println("Enter three points for a triangle:");  
 x1 = input.nextDouble();  
 y1 = input.nextDouble();  
 x2 = input.nextDouble();  
 y2 = input.nextDouble();  
 x3 = input.nextDouble();  
 y3 = input.nextDouble();  
  
 *//计算三边长* s1 = Math.*sqrt*(Math.*pow*(x1 - x2, 2) + (Math.*pow*(y1 - y2, 2)));  
 s2 = Math.*sqrt*(Math.*pow*(x1 - x3, 2) + (Math.*pow*(y1 - y3, 2)));  
 s3 = Math.*sqrt*(Math.*pow*(x3 - x2, 2) + (Math.*pow*(y3 - y2, 2)));  
  
 *//判断三角形是否符合定义* if (s1 + s2 <= s3 || Math.*abs*(s1 - s2) >= s3 || s3 + s2 <= s1 || Math.*abs*(s3 - s2) >= s1 || s3 + s1 <= s2 || Math.*abs*(s3 - s1) >= s2)  
 System.*out*.println("Invalid Triangle !");  
 else {  
 s = (s1 + s2 + s3) / 2.0;  
 System.*out*.println("The area of the triangle is " + Math.*sqrt*(s \* (s - s1) \* (s - s2) \* (s - s3)));  
 }  
 }  
}

源代码1.3 Exp\_3.java

public class Exp\_3 {  
 public static void main(String[] args) {  
 *//产生随机数* int rand\_num = (int) (Math.*random*() \* 52) + 1;  
 String suit;  
  
 *//判断花色* if (rand\_num >= 1 && rand\_num <= 13)  
 suit = "Clubs";  
 else if (rand\_num <= 26)  
 suit = "Diamond";  
 else if (rand\_num <= 39)  
 suit = "Heart";  
 else  
 suit = "Spades";  
  
 String weight = "0";  
  
 *//判断大小* if (rand\_num % 13 >= 2 && rand\_num % 13 <= 10)  
 weight = String.*valueOf*(rand\_num % 13);  
 if (rand\_num % 13 == 1)  
 weight = "Ace";  
 if (rand\_num % 13 == 11)  
 weight = "Jack";  
 if (rand\_num % 13 == 12)  
 weight = "Queen";  
 if (rand\_num % 13 == 0)  
 weight = "King";  
  
 *//输出结果* System.out.println("The random number is " + rand\_num);  
 System.out.println("The chard generated just now is " + weight + " in " + suit);  
 }  
}

源代码1.4 Exp\_4.java

public class Exp\_4 {  
 public static void main(String[] args) {  
 printNum();  
 }  
  
 *//判断传入的数值是否符合要求* public static boolean isValid(int i){  
 return (i % 5 == 0 || i % 6 == 0) && !(i % 5 == 0 && i % 6 == 0);  
 }  
  
 *//打印符合要求的数字* public static void printNum(){  
 byte valid\_case = 0;  
 for (int i = 100;i<=200;i++){  
 if (isValid(i)){  
 valid\_case++;  
 System.out.print(i+"\t");  
 }  
 else  
 continue;  
 if (valid\_case%10==0)*//每输出10个数，换一次行* System.out.print('\n');  
 }  
 }  
}