

**学生实验报告**

|  |  |
| --- | --- |
| 课程名称 | Java程序设计 |
| 学院 | 数学与计算机科学学院 |
| 学科专业 | 计算机科学与技术 |
| 班级 | 计师171 |
| 学号 | 201730210423 |
| 姓名 | 韩 玮 |
| 成绩 |  |
| 教师 | 宋炯老师 |

目 录

[第三章 2](#_Toc29505)

[第四章 4](#_Toc26710)

[第七章 6](#_Toc17256)

[第八章 10](#_Toc7858)

[第九章 11](#_Toc13573)

[第十二章 19](#_Toc28777)

[第十三章 26](#_Toc7766)

[第十五章 30](#_Toc8885)

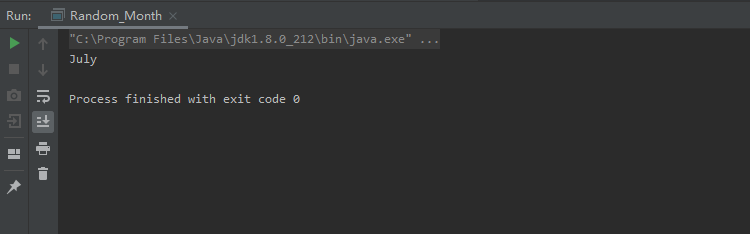
## 第三章

3.4 编写一个随机产生1和12之间整数的程序，并且根据数字1，2，…，12显示相应的英文月份：January，February，…，December。

程序代码：

|  |
| --- |
| package Chap3;  import java.util.\*;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 16:25  \** ***@Version*** *1.0  \*\*/* public class Random\_Month {  public static void main(String[] args) {  int m = 1+(int)(Math.*random*()\*12);  switch (m){  case 1: System.*out*.println("January");  break;  case 2: System.*out*.println("February");  break;  case 3: System.*out*.println("March");  break;  case 4:System.*out*.println("Apirl");  break;  case 5:System.*out*.println("May");  break;  case 6:System.*out*.println("June");  break;  case 7:System.*out*.println("July");  break;  case 8:System.*out*.println("August");  break;  case 9:System.*out*.println("September");  break;  case 10:System.*out*.println("October");  break;  case 11:System.*out*.println("November");  break;  default:System.*out*.println("December");  }  } } |

运行截图：

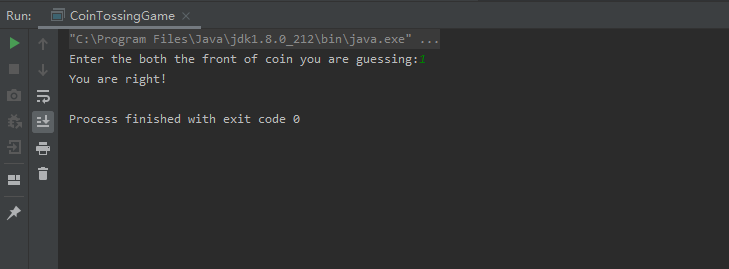


3.14（游戏：猜硬币的正反面）编写程序，让用户猜一猜是硬币的正面还是反面。这个程序随机产生一个整数0或者1，它们分别表示硬币的正面和反面。程序提示用户输入一个猜测值，然后报告这个猜测值是正确的还是错误的。

程序代码：

|  |
| --- |
| package Chap3;  import java.util.\*; */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 16:42  \** ***@Version*** *1.0  \*\*/* public class CoinTossingGame {  public static void main(String[] args){  Scanner input = new Scanner(System.*in*);   int coin=(int)(Math.*random*()\*2);   System.*out*.print("Enter the both the front of coin you are guessing:");  int guess = input.nextInt();   if (guess==coin)  System.*out*.println("You are right!");  else  System.*out*.println("You are wrong!");  } } |

运行截图：



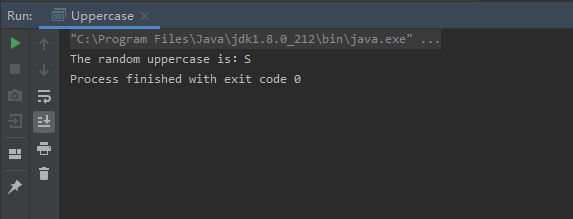
## 第四章

4.16 编写一个程序，使用Math.random()方法显示一个随机的大写字母

程序代码：

|  |
| --- |
| package Chap4;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 16:54  \** ***@Version*** *1.0  \*\*/* public class Uppercase {  public static void main(String[] args){  int i=(int)(Math.*random*()\*26);  char ch=(char)(Math.*random*()\*26+'A');  System.*out*.print("The random uppercase is：" + ch);  } } |

运行截图：

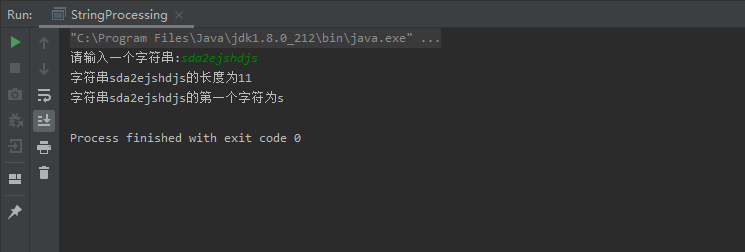


4.20 （字符串处理）编写一个程序，提示用户输入一个字符串，显示它的长度和第一个字母。

程序代码：

|  |
| --- |
| package Chap4;  import java.util.\*; import java.util.stream.Stream;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 17:00  \** ***@Version*** *1.0  \*\*/* public class StringProcessing {  public static void main(String[] args){  System.*out*.print("请输入一个字符串:");  Scanner input = new Scanner(System.*in*);  String s = input.next();  System.*out*.println("字符串"+s+"的长度为"+s.length());  System.*out*.println("字符串"+s+"的第一个字符为"+s.charAt(0));  } } |

运行截图：

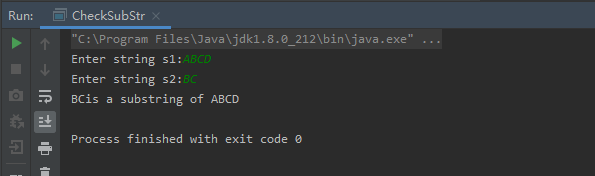


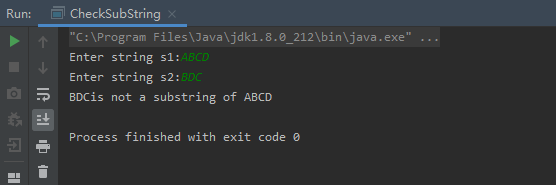
4.22 (检测子串）编写一个程序，提示用户输入两个字符串，检测第二个字符串是否是第一个字符串的子串。

程序代码：

|  |
| --- |
| package Chap4;  import java.util.\*; */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 20:21  \** ***@Version*** *1.0  \*\*/* public class CheckSubString {  public static void main(String[] args){  System.*out*.print("Enter string s1:");  Scanner input = new Scanner(System.*in*);  String s1 = input.next();  System.*out*.print("Enter string s2:");  String s2 = input.next();  if (s1.contains(s2))  System.*out*.println(s2 + "is a substring of " + s1);  else  System.*out*.println(s2 + "is not a substring of " + s1);  } } |

运行截图：





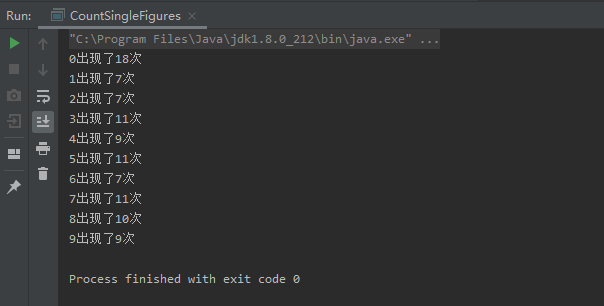
## 第七章

7.7 编写一个程序，生成0和9之间的100个随机整数，然后显示每一个数出现的次数。

程序代码：

|  |
| --- |
| package Chap7;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 20:34  \** ***@Version*** *1.0  \*\*/* public class CountSingleFigures {  public static void main(String[] args){  int[] counts = new int[10];  for (int i = 0; i < 100 ; i++){  counts[(int)(Math.*random*()\*10)] ++;  }  for (int i = 0;i < 10 ; i++){  System.*out*.println(i + "出现了" + counts[i] + "次");  }  } } |

运行截图：

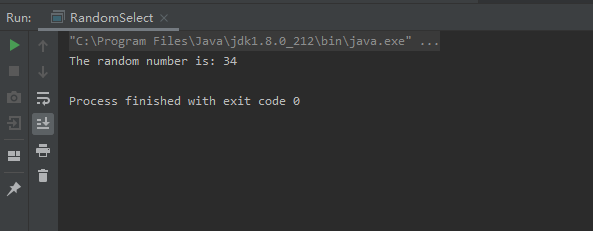


7.13 （随机数选择器）编写一个方法,返回1到54之间的随机数,但不能是传递到实参中的数。

程序代码：

|  |
| --- |
| package Chap7;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 20:51  \** ***@Version*** *1.0  \*\*/* public class RandomSelect {  public static int getRandom(int number){ //可变长参数列表  return (int)(Math.*random*()\*number);  }  public static void main(String[] args){  System.*out*.println("The random number is: " + *getRandom*(55));  } } |

运行截图：

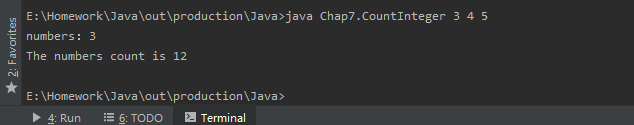


7.21（整数求和）编写程序，从命令行输入不定数目的整数，然后显示它们的和。

程序代码：

|  |
| --- |
| package Chap7;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 21:11  \** ***@Version*** *1.0  \*\*/* public class CountInteger {  public static void main(String[] args){  int sum=0;  System.*out*.println("numbers："+args.length);  for(String arg:args) {  sum += Integer.*parseInt*(arg); //将字符串类型转化为整型并求和  }  System.*out*.println("The numbers count is " + sum);  } } |

运行截图：

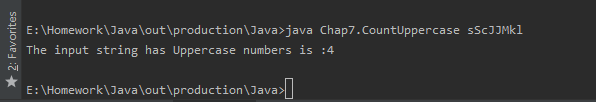


7.22 （计算一个字符串中大写字母的数目）java编写程序，从命令行输入一个字符串，然后显示字符串中大写字母的数目

程序代码：

|  |
| --- |
| package Chap7;  import java.util.\*; */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 21:20  \** ***@Version*** *1.0  \*\*/* public class CountUppercase {  public static void main(String[] args){  String s = args[0];  char[] arr = s.toCharArray();  int sum = 0;//统计多少个大写字母  for(int i = 0; i < arr.length; i++){  if((int)arr[i] >= 65 && (int)arr[i] <= 90){  sum = ++sum;  }  }  System.*out*.println("The input string has Uppercase numbers is :" + sum);  } } |

运行截图：



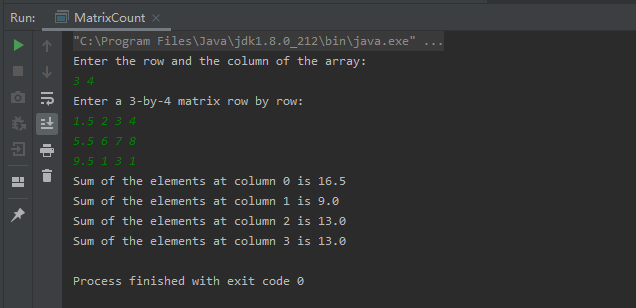
## 第八章

8.1（求矩阵中各列数字的和）编写一个方法，求整数矩阵中特定列的所有元素的和。

程序代码：

|  |
| --- |
| package Chap8;  import java.util.\*; */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/9 21:32  \** ***@Version*** *1.0  \*\*/* public class MatrixCount {  public static double sumColumn(double[][] m,int columnIndex)  {  double sum=0;  for(int i=0;i<m.length;i++){  sum=sum+m[i][columnIndex];  }  return sum;  }  public static void main(String[] args)  {  int m,n;  double s;  double[][] a;  Scanner input=new Scanner(System.*in*);  System.*out*.println("Enter the row and the column of the array: ");  m=input.nextInt();  n=input.nextInt();  System.*out*.println("Enter a "+m+"-by-"+n+" matrix row by row:");  a=new double[m][n];  for(int row=0;row<a.length;row++){  for(int column=0;column<a[row].length;column++){  a[row][column]=input.nextDouble();  }  }  for(int i=0;i<n;i++){  s=*sumColumn*(a,i);  System.*out*.println("Sum of the elements at column "+i+" is "+s);  }  } } |

运行截图：



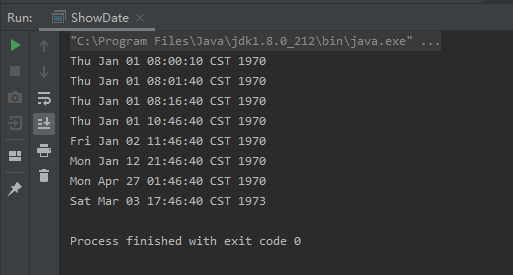
## 第九章

9.3（使用日期类 Date）编写程序创建一个 Date 对象，设置它的流逝时间分别为 10000、100000、1000000、10000000、100000000、1000000000、10000000000、100000000000 ，然后使用 toString() 方法分别显示上述日期。

程序代码：

|  |
| --- |
| package Chap9;  import java.util.\*; */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/10 8:42  \** ***@Version*** *1.0  \*\*/* public class ShowDate {  public static void main(String[] args){  Date date = new Date();  int count = 0;  for (long l = 10000;count < 8 ; count ++ , l\*=10){  date.setTime(l);  System.*out*.println(date.toString());  }  } } |

运行截图：

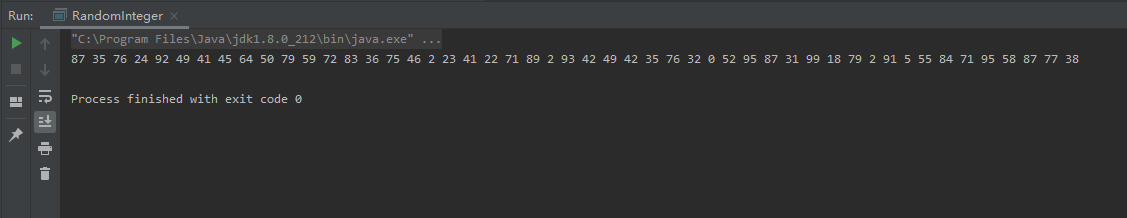


9.4（使用随机类Random）编写一个程序，创建种子是1000的Rnadom对象，然后使用nextInt(100)方法显示0到100之间前50个随机整数。

程序代码：

|  |
| --- |
| package Chap9;  import java.util.\*; */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/10 8:50  \** ***@Version*** *1.0  \*\*/* public class RandomInteger {  public static void main(String[] args){  Random random = new Random(1000);  for (int i = 0;i < 50 ; i++)  System.*out*.print(random.nextInt(100) + " ");  System.*out*.print("\n");  } } |

运行截图：



9.7设计一个名为Account类，它包括：

一个名为id的int类型私有帐户数据域（默认值0）。

一个名为balance的double类型私有帐户数据域（默认值0）。

一个名为annualInterestRate的double类型私有数据域存储当前利率（默认值0）。假设所以的帐户都有相同的利率。

一个名为dateCreated的Date类型私有数据域存储帐户的开户日期。

一个能创建默认帐户的无参构造方法。

一个能创建带特定id和初始余额的帐户的构造方法。

Id balance annualInterstRate的访问器和修改器。

dateCreated的访问器。一个名为getMonthlyInterestRate()的方法返回月利率。

一个名为withDraw的方法从帐户提取特定数额。

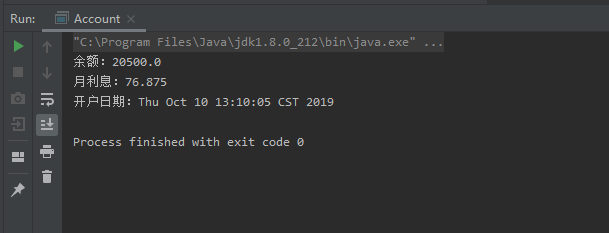
一个名deposit的方法向帐户存储特定的数额。

画出该类的UML图。实现这个类。编写一个测试程序，创建一个帐户ID为1122，余额为20000美元，年利率为4.5%的Account对象。使用withdraw方法取款2500美元，使用deposit方法存款3000美元，然后打印余额，月利息以及这个帐户的开户日期。

代码程序：

|  |
| --- |
| package Chap9;  import java.util.\*; */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/10 8:54  \** ***@Version*** *1.0  \*\*/* public class Account {  private int id; //账户号码  private double balance; //账户余额  private double annualInterestRate; //当前利率  Date dateCreated = new Date();  public Account()  {  this.id = 0;  this.balance = 0;  this.annualInterestRate = 0;  }  public Account(int id,int balance)  {  this.id = id;  this.balance = balance;  }  public int getId()  {  return this.id;  }  public void steId(int id)  {  this.id = id;  }  public double getBalance()  {  return this.balance;  }  public void setBalance(double balance)  {  this.balance = balance;  }  public double getAnnualInterestRate()  {  return this.annualInterestRate;  }  public void setAnnualInterestRate(double annualInterestRate)  {  this.annualInterestRate = annualInterestRate/100;  }  public Date DateCreated()  {  return this.dateCreated;  }  public double getMonthlyInterestRate()  {  return this.annualInterestRate/12;  }  public double getMonthlyInterest()  {  return balance\*getMonthlyInterestRate();  }  public double withDraw(double count)  {  this.balance -= count;  return this.balance;  }  public double deposit(double count)  {  this.balance += count;  return this.balance;  }  public static void main(String[] args)  {  Account account = new Account(1122,20000);  account.setAnnualInterestRate(4.5);  account.withDraw(2500);  account.deposit(3000);  System.*out*.println("余额：" + account.getBalance());  System.*out*.println("月利息：" + account.getMonthlyInterest());  System.*out*.println("开户日期："+account.DateCreated().toString());  } } |

运行截图：



9.8（Fan类）设计一个名为Fan的类来表示一个风扇。这个类包括：

三个名为SLOW、MEDIUM和FAST而值是1、2和3的常量表示风扇的速度。

一个名为speed的int类型私有数据域表示风扇的速度（默认值SLOW）。

一个人名为on的boolean类型私有数据域表示风扇是否打开（默认值为false）。

一个名为radius的double类型私有数据域表示风扇的半径（默认值5）。

一个名为color的string类型数据域表示风扇的颜色（默认值为blue）。

这四个数据域的访问器和修改器。

一个创建默认风扇的无参构造方法。

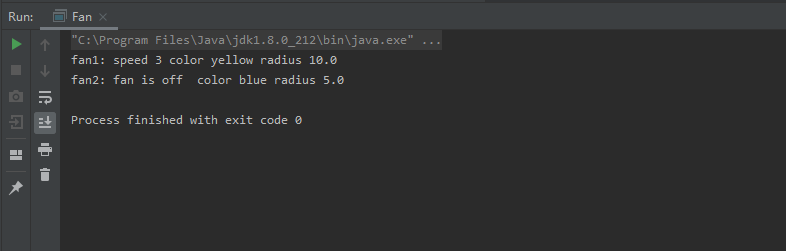
一个名为toString()方法返回描述风扇的字符串。如果风扇是打开的，那么该方法在一个组合的字符串中返回风扇的速度、颜色和半径。如果风扇没有打开，该方法就会返回一个由“fan is off”和风扇颜色及半径组合成的字符串。

画出该类的UML图。实现这个类。编写一个测试程序，创建两个Fan对象。将第一个对象设置为最大速度、半径10、颜色为yellow、状态为打开。将第二个对象设置为中等速度、半径为5、颜色为blue、状态为关闭。通过调用它们的toSrting方法显示这些对象。

程序代码：

|  |
| --- |
| package Chap9;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/7 14:38  \** ***@Version*** *1.0  \*\*/* public class Fan {  final static int *SLOW* = 1;  final static int *MEDIUM* = 2;  final static int *FAST* = 3;  private int speed = *SLOW*;  private boolean on = false;  private double radius = 5.0;  public String color = "blue";   public void setSpeed(int newSpeed)  {  this.speed = newSpeed;  }  public int getSpeed()  {  return speed;  }   public void setOn(boolean newState)  {  this.on = newState;  }  public boolean getOn()  {  return on;  }   public void setRadius(double newRadius)  {  this.radius = newRadius;  }  public double getRadius()  {  return radius;  }   public void setColor(String newColor)  {  this.color = newColor;  }  public String getColor()  {  return color;  }   public Fan(){  }   public String toString()  {  if(on)  {  return " speed " + speed + " color " + color + " radius " + radius;  }  else  {  return " fan is off " + " color " + color + " radius " + radius;  }  }  public static void main(String[] args){  Fan fan1 = new Fan();  Fan fan2 = new Fan();  fan1.setSpeed(*FAST*);  fan1.setRadius(10);  fan1.setColor("yellow");  fan1.setOn(true);   fan2.setSpeed(*MEDIUM*);  fan2.setRadius(5);  fan2.setColor("blue");  fan2.setOn(false);   System.*out*.println("fan1:" + fan1.toString());  System.*out*.println("fan2:" + fan2.toString());  } } |

运行截图：



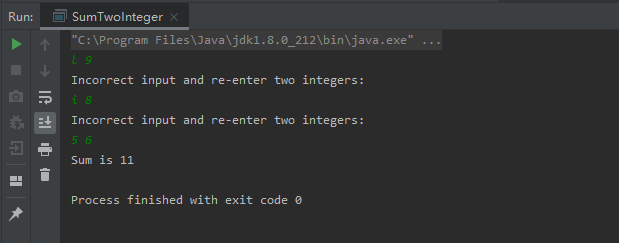
## 第十二章

12.2(NumberFormatException数字格式异常)编写一个程序，提示用户读取两个整数，然后显示他们的和。程序应该在输入不正确时提示用户再次输入数字。

程序代码：

|  |
| --- |
| package Chap12;  import java.util.\*;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/22 10:18  \** ***@Version*** *1.0  \*\*/* public class SumTwoInteger {  public static void main(String[]args)  {  Scanner input = new Scanner(System.*in*);  int a,b;  while(true)  {  try {  a = input.nextInt();  b = input.nextInt();  System.*out*.println("Sum is "+(a+b));  }  catch (InputMismatchException e)  {  input.nextLine();  System.*out*.println("Incorrect input and re-enter two integers:");  continue;  }  break;  }  } } |

运行截图：



12.3（ArrayIndexOutBoundsException异常）编写一个程序

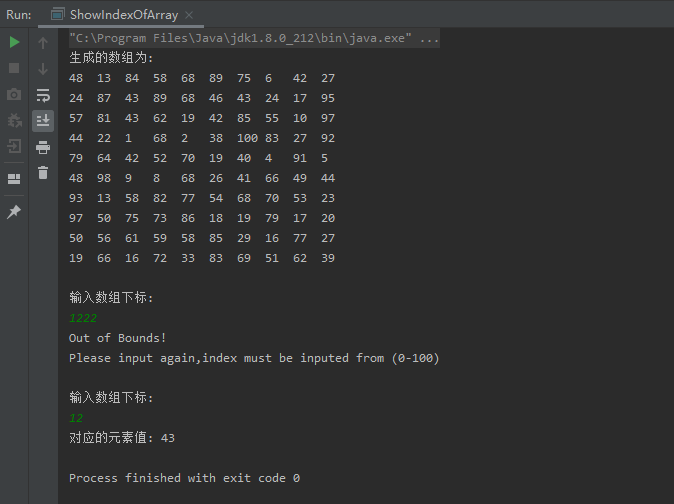
1.创建一个由100个随机选取的整数构成的数组

2.提示用户输入数组的下标，然后显示对应的元素值。如果指定下标越界，就显示“Out Of Bound”

程序代码：

|  |
| --- |
| package Chap12;  import java.util.\*;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/22 12:56  \** ***@Version*** *1.0  \*\*/* public class ShowIndexOfArray {  public static void main(String[]args)  {  int[] array = new int[100];  int count = 0;  for(int i = 0; i < 100 ; i++)  {  array[i] = ((int)(Math.*random*()\*100) + 1);  }  System.*out*.println("生成的数组为: ");  for (int i = 0 ; i < array.length;i++)  {  System.*out*.print(array[i] + "\t");  count ++ ;  if (count % 10 == 0 && count > 0)  {  System.*out*.println();  }  }  while (true)  {  try {  Scanner input = new Scanner(System.*in*);  System.*out*.println("\n输入数组下标: ");  int index = input.nextInt();  System.*out*.println("对应的元素值: " + array[index]);  } catch (ArrayIndexOutOfBoundsException ex){  System.*out*.println("Out of Bounds!");  System.*out*.println("Please input again,index must be inputed from (0-100)");  continue;  }  break;  }   } } |

运行截图：

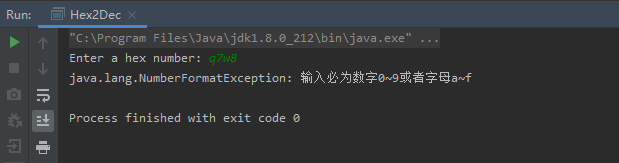


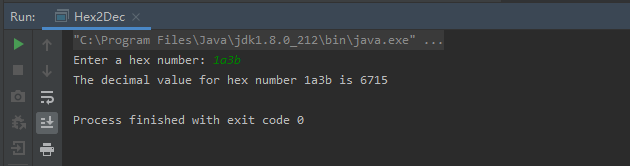
12.6（Numberformatexception异常）程序清单6-8实现了hexToDec（String hexString）方法,它将一个十六进制字符串转换为一个十进制数。实现这个hexToDec（）方法，在字符串不是一个十六进制字符串时抛出NumberFormat

程序代码：

|  |
| --- |
| package Chap12;  import java.util.Scanner;  public class Hex2Dec {  */\*\* Main method \*/* public static void main(String[] args) {  // Create a Scanner  Scanner input = new Scanner(System.*in*);   // Prompt the user to enter a string  System.*out*.print("Enter a hex number: ");  try {  String hex = input.nextLine();   System.*out*.println("The decimal value for hex number "  + hex + " is " + *hexToDec*(hex.toUpperCase()));  } catch (NumberFormatException ex){  System.*out*.println(ex.toString());  }   }   public static int hexToDec(String hexString) throws NumberFormatException{  int decimalValue = 0;  for (int i = 0; i < hexString.length(); i++) {  char hexChar = hexString.charAt(i);  if ((hexChar>='A'&& hexChar <='F') || (hexChar>='0'&&hexChar<='9'))  {  decimalValue = decimalValue \* 16 + *hexCharToDecimal*(hexChar);  }  else  throw new NumberFormatException("输入必为数字0~9或者字母a~f");  }  return decimalValue;  }   public static int hexCharToDecimal(char ch) {  if (ch >= 'A' && ch <= 'F')  return 10 + ch - 'A';  else // ch is '0', '1', ..., or '9'  return ch - '0';  } } |

运行截图：



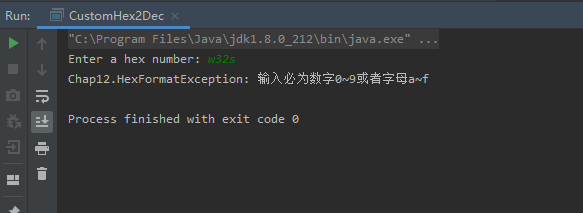


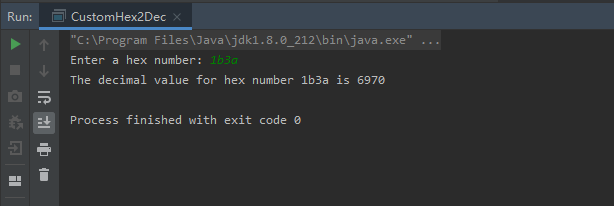
12.8( HexFormatException 异常）编程练习题 12.6 实现 hex2Dec 方法，在字符串不是一个十六进 制字符串时抛出 NumberFormatException 异常。定义一个名为 HexFormatException的自定 义异常。实现 hex2Dec 方法，在字符串不是一个十六进制字符串时抛出 HexFormatException 异常。

程序代码：

|  |
| --- |
| package Chap12;  import java.util.Scanner;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/10/22 21:28  \** ***@Version*** *1.0  \*\*/* class HexFormatException extends NumberFormatException {  public HexFormatException(){  super("输入必为数字0~9或者字母a~f");  } } public class CustomHex2Dec {  */\*\* Main method \*/* public static void main(String[] args) {  // Create a Scanner  Scanner input = new Scanner(System.*in*);   // Prompt the user to enter a string  System.*out*.print("Enter a hex number: ");  try {  String hex = input.nextLine();   System.*out*.println("The decimal value for hex number "  + hex + " is " + *hexToDec*(hex.toUpperCase()));  } catch (HexFormatException ex){  System.*out*.println(ex.toString());  }   }   public static int hexToDec(String hexString) throws HexFormatException{  int decimalValue = 0;  for (int i = 0; i < hexString.length(); i++) {  char hexChar = hexString.charAt(i);  if ((hexChar>='A'&& hexChar <='F') || (hexChar>='0'&&hexChar<='9'))  {  decimalValue = decimalValue \* 16 + *hexCharToDecimal*(hexChar);  }  else  throw new HexFormatException();  }  return decimalValue;  }   public static int hexCharToDecimal(char ch) {  if (ch >= 'A' && ch <= 'F')  return 10 + ch - 'A';  else // ch is '0', '1', ..., or '9'  return ch - '0';  } } |

运行截图：





## 第十三章

1.设计一个动物声音“模拟器”，希望模拟器可以模拟许多动物的叫声

1.编写抽象类Animal

2.编写模拟器类Simulator

3.编写Animal类的子类：Dog和Cat类

4.编写主类Application（用户程序）

包含代码：Simulator simulator=new Simulator();

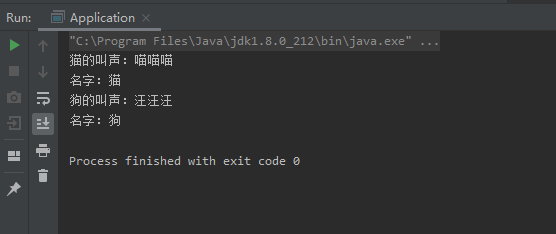
simulator.playSound(new Cat());

simulator.playSound(new Dog());

程序代码：

|  |
| --- |
| package Chap13;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/11/1 19:48  \** ***@Version*** *1.0  \*\*/* interface Animal {  public void cry();  public String getAnimaName(); }  class Dog implements Animal{  public void cry(){  System.*out*.println("狗的叫声：汪汪汪");  }  public String getAnimaName(){  return "名字：狗";  } }  class Cat implements Animal{  public void cry(){  System.*out*.println("猫的叫声：喵喵喵");  }  public String getAnimaName(){  return "名字：猫";  } }  class Simulator{  void playSound(Animal animal){  animal.cry();  System.*out*.println(animal.getAnimaName());  } } public class Application {  public static void main(String[] args){  Simulator simulator = new Simulator();  simulator.playSound(new Cat());  simulator.playSound(new Dog());  } } |

运行截图：

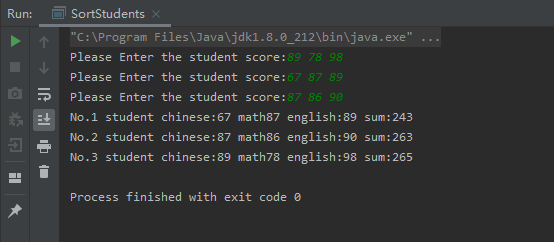


2.定义一个Student类，其中有x, y, z, sum字段，分别表示其语文、数学、英语成绩和三门课总分。要求该类实现Comparable接口，对其进行排序时按照总分从高到低的次序排序。写一个Test程序输入三个学生的三门课成绩并排序后输出。

程序代码：

|  |
| --- |
| package Chap13;  import java.util.\*; */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/11/1 20:24  \** ***@Version*** *1.0  \*\*/* class Student implements Comparable<Student>{  private int x;  private int y;  private int z;  public int sum;  Student(int x,int y,int z){  this.x = x;  this.y = y;  this.z = z;  this.sum = x + y + z;  }   @Override  public int compareTo(Student o) {  if (sum > o.sum)  return 1;  else if (sum < o.sum)  return -1;  else  return 0;  }   @Override  public String toString() {  return "chinese:"+this.x+" math"+this.y+" english:"+this.z+" sum:"+this.sum;  } } public class SortStudents {  public static void main(String[] args){  int x,y,z;  Scanner input = new Scanner(System.*in*);   Student[] students = new Student[3];  for(int i = 0 ;i < 3 ; i++){  System.*out*.print("Please Enter the student score:");  x = input.nextInt();  y = input.nextInt();  z = input.nextInt();  students[i] = new Student(x,y,z);  }   Arrays.*sort*(students);  int i = 1;  for (Student student:students){  System.*out*.println("No." + i +" student " + student.toString());  i++;  }  } } |

运行截图：



## 第十五章

1. (创建一个简单的计算器)编写一个程序完成加法、减法、乘法和除法操作

程序代码：

|  |
| --- |
| Calculator.class  package Chap15;  import javax.swing.\*;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/11/3 16:03  \** ***@Version*** *1.0  \*\*/* public class Calculator extends JFrame {  public static void main(String args[]) {  WindowBoxLayout win = new WindowBoxLayout();  win.setBounds(100,100,550,120);  } }  WindowBoxLayout.class  package Chap15;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener;  */\*\*  \** ***@Author*** *Lan XiaoJun  \** ***@Date*** *2019/11/3 18:37  \** ***@Version*** *1.0  \*\*/* public class WindowBoxLayout extends JFrame {  Box boxOne,boxTwo; //行式盒  Box boxV; //列式盒  public WindowBoxLayout(){  setLayout(new FlowLayout());  init();  setVisible(true);  setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  }  void init(){  boxOne = Box.*createHorizontalBox*();  boxTwo = Box.*createHorizontalBox*();  boxV = Box.*createVerticalBox*();  JTextField numOne = new JTextField(10);  JTextField numTwo = new JTextField(10);  JTextField result = new JTextField(10);  JButton add = new JButton("Add");  JButton sub = new JButton("Subtract");  JButton mul = new JButton("Multiply");  JButton div = new JButton("Divide");  boxOne.add(new JLabel("Number 1:"));  boxOne.add(numOne);  boxOne.add(new JLabel("Number 2:"));  boxOne.add(numTwo);  boxOne.add(new JLabel("Result:"));  boxOne.add(result);  boxTwo.add(add);  boxTwo.add(sub);  boxTwo.add(mul);  boxTwo.add(div);  boxV.add(boxOne);  boxV.add(Box.*createVerticalStrut*(10));  boxV.add(boxTwo);  add(boxV);  add.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  try{  double num1= Double.*parseDouble*(numOne.getText().trim());  double num2= Double.*parseDouble*(numTwo.getText().trim());  result.setText(num1+num2+"");  }  catch (Exception ex)  {  numOne.setText("");  numTwo.setText("");  }  }  });  sub.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  try {  double num1= Double.*parseDouble*(numOne.getText().trim());  double num2= Double.*parseDouble*(numTwo.getText().trim());  result.setText(num1-num2+"");  }  catch (Exception ex)  {  numOne.setText("");  numTwo.setText("");  }  }  });  mul.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  try {  double num1= Double.*parseDouble*(numOne.getText().trim());  double num2= Double.*parseDouble*(numTwo.getText().trim());  result.setText(num1\*num2+"");  }  catch (Exception ex)  {  numOne.setText("");  numTwo.setText("");  }  }  });  div.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  try {  double num1= Double.*parseDouble*(numOne.getText().trim());  double num2= Double.*parseDouble*(numTwo.getText().trim());  result.setText(num1/num2+"");  }  catch (Exception ex)  {  numOne.setText("");  numTwo.setText("");  }  }  });  } } |

运行截图：

