第 3 章

类的方法

要点导读

本章介绍类的方法。Java 中的流程控制结构主要有顺序结构、选择结构及循环结构三种。顺序结构即是按照从上到下的顺序执行语句，没有转移及重复。选择结构是根据给定的条件成立与否，执行不同的语句或语句组。Java 的选择结构主要有 if 选择结构及 switch 选择结构两种。值得注意的是，使用 switch 选择结构时，最容易犯的错误是漏写 break 语句。循环控制结构是在一定的条件下，反复执行某段程序的流程结构，被反复执行的程序称为循环体。Java 中提供的循环语句共有三种：for 语句、while 语句、do-while 语句。

异常(Exception)又称为例外，是特殊的运行错误对象。Java 提供了异常处理机制，在 Exception 类中定义了程序产生异常的条件。对待异常通常并不是简单地结束程序，而是转去执行某段特殊代码处理这个异常，设法恢复程序继续执行。Java 中预定义的常用异常有 ArithmeticException、NullPointerException、ArrayIndexOutOfBoundsExceptionJava、NegativeArraySizeException、FileNotFoundException、IOException 等。对异常的处理，可以声明抛出异常，由调用方法处理，也可以捕获并处理。抛出异常使用 throws 字句实现，捕获异常使用 try、catch 和 finally 语句捕获和处理。程序员也可以自定义异常类，表示某类异常。

方法的重载(overloading)是指在一个类中可以有名字相同的多个方法，但这些方法的参数必须不同，或者是参数个数不同，或者是参数类型不同，但返回值可以相同。

从 Java 6 开始，可以在方法中执行脚本。Java 的脚本 API 提供了一个独立于各种脚本语言的框架，利用该框架可以在 Java 代码中执行脚本。

实验 3　类的方法

1. 实验目的

(1) 掌握三种流程控制语法，并熟练应用。

(2) 了解 Java 的异常处理机制，会编写相应程序。

(3) 掌握方法重载的含义，并熟练应用。

2. 实验任务

(1) 复习、熟悉流程控制的语法。

目测,笔算出下面程序段的执行结果。

```
public static void main(String[] args) {
        int result=0;
        int j=-100;
        for (int i=0; i<100; i++) {
            if (i %3==0) {
                j=i;
            } else {
                j=i * 2;
                while (j>0) {
                    j--;
                    result+=j;
                }
            }
            if (i %3==1) {
                result--;
                continue;
            }
            result+=i;
            if (result>0) {
                break;
            }
        }
        System.out.println(result);
    }
```

(2) 函数返回及异常处理。

目测以下程序代码能编译通过吗? 不能的话有什么错误? 若有错误则尝试在不改变原程序语义的情况下,给出两种解决方案并分别实现在 Calculator1.java 和 Calculator2.java 中(两种方法针对 Exception 的错误)。(找错误的时候先不使用 IDE 辅助,尽可能的找出所有错误)。

```
public class Calculator {

    public float getValue(String type) throws Exception {
        //获得产品单价
        if (type.equals("cookie")) {
            return 1.11f;
        } else if (type.equals("pie")) {
            return 5.5f;
        }
```

```
    }

    public int getValue(String type) throws Exception {
        //获得产品数量
        if (type.equals("cookie")) {
            return 10;
        } else {
            return 20;
        }
    }

    public float calculate() {
        float price=getValue("cookie");
        int amount=getValue("cookie");
        return price * amount;
    }
}
```

(3) 随机数的生成。

实现一个方法生成 20 个 1～100 的随机整数。每生成一个计算当前已生成随机数的平均值,并打印一行信息,包括当前随机数值,比前一个随机数大还是小(Greater/Less/Equal),当前的平均值,当前随机数比当前平均值大还是小。

样例输出:

```
[Value]     [G/L than Prev.]    [Avg.]     [G/L than Avg.]
-------------------------------------------------------------
25          ---                 25.0       Equal
70          Greater             47.5       Greater
30          Less                41.7       Less
10          Less                33.8       Less
⋮           ⋮                   ⋮          ⋮
```

(4) 商品找零。

从键盘上输入一件商品价格(0.01～5.00 元)。如果支付 5 元,给出一种找零方案,使得所找纸币及硬币的个数最少。例如,输入物品价格 1.68 元,给出找零方案 2 元 1 张,1 元 1 张,2 角 1 个,1 角 1 个,2 分 1 个。(可找币种类包括所有发行的纸币和硬币)。

另:如果输入 0.01～5.00 则给出方案并继续等待下个输入;如果输入 0,则程序退出。如果输出 3 位(含)以上的小数,例如 3.618 则抛出自定义异常 WrongFormatException,并让用户继续重新输入;如果输入负数或者大于 5 的数值,则抛出自定义异常 InputOutOfRangeException,并且程序终止。

3. 实验步骤

(1) 复习控制流程语法。

① 习惯使用大括号标明程序段。养成良好的编程习惯,即使分支、循环中只有一行

代码，也最好用大括号括起来。

② 按照缩进格式书写的代码更能清晰地看出程序流程的结构。右键单击打开的代码。选择“重新格式化代码”(组合键 Ctrl＋Shift＋F)可以让 netbeans 帮助格式化代码。相关格式的修改可以在“选项”→“编辑器”对话框中修改。

③ 目测题目中的代码片断，计算出程序输出。

④ 如果完毕后，可以将程序输入 NetBeans 以验证自己的计算。

(2) 解决实验题 2 中的编译错误。

① 目测题目中代码的编译错误。指出这些错误，并想出两种异常的处理方法。

② 将两种解决方案实现。

(3) 在 NetBeans 中创建新的 Java 项目 Exp3。

① 选择“文件”→“新建项目”，在“类别”下选择 Java，在“项目”下选择“Java 应用程序”，然后单击“下一步”按钮。

② 在“项目名称”下输入 Exp3，不要勾选“创建主类”选项。

③ 养成良好的命名习惯。包名使用小写字母开头。类名使用大写字母开头。

(4) 验证实验题 2。

① 创建 java 包，名称为“problem2”。

② 在“problem2”包下创建 Calculator. java。将题目中的代码输入，并编译。看看自己是否找到了所有的错误(编译器只显示最高层的错误，将高层的错误修改后可以看到更细节的错误)。实际应该找到 3 个不同类型的编译错误。

③ 在“problem2”包下创建 Calculator1. java 和 Calculator2. java。在不大量改变原程序语义的情况下，修改 Calculator. java 中的错误。分别将两种修改方案在这两个类中实现。

(5) 实现实验题 3 中的方法。

① 在项目 Exp3 中创建新的包和类。创建 java 包，名称为 problem3。在 problem3 包下创建 RandomGenerator. java。

② 实现题目中要求的方法，并执行验证。

③ 学习使用 java. util. Random 类来生成随机数。

注意：*养成良好的命名习惯，方法名使用小写字母开头。*

(6) 实现题目 4 中的类。

① 在项目 Exp3 中创建新的包和类。创建 java 包，名称为 problem4。在 problem4 包下创建 ChangeCalculator. java。

② 创建 WrongFormatException. java，InputOutOfRangeException. java。并实现这两个自定义异常类。

③ 在 ChangeCalculator 类中实现题目中要求的方法。

④ 对程序进行验证，输入以下数值验证程序的输出。

```
输入：0          输出：程序退出
输入：5.00       输出：无需找零的方案
输入：1.68       输出：2元1张,1元1张,2角1个,1角1个,2分1个的方案
```

输入：0.01　　输出：2 元 2 张,5 角 1 个,2 角 2 个,5 分 1 个,2 分 2 个的方案
输入：2.222　　输出：显示异常 WrongFormatException 并等待重新输入
输入：6.05　　输出：显示异常 InputOutOfRangeException 并退出

注意：此题最容易出现的错误是少找一分钱,例如,输入 0.01 时,输出的不是 2 个 2 分,而是 1 个 2 分和 1 个 1 分。其原因在于浮点数在计算机中的表示是不精确的。此题中,如果要知道某浮点数对应的整数,可以使该浮点数加上一个很小的数,并对结果取整。例如：对浮点数 8.99,如果想得到结果 9,可以使 8.99 加上 0.02,再取整,就可以得到整数 9,即：

```
float f=8.99;
int i=(int)(f+0.02);
```

此时 i 的值为 9。

习题解答

3-1　设 n 为自然数,

$$n!=1\times2\times3\times\cdots\times n$$

称为 n 的阶乘,并且规定 0!=1。试编程计算 2!,4!,6! 和 10!,并将结果输出到屏幕上。

解：新建 Exe3_1.java 文件,其内容为：

```
class Factorial {
    public static int calcFactorial(int n) {
        if (n==0)
            return 0;
        else {
            int ret=1;
            for(int i=1; i<=n; i++) {
                ret*=i;
            }
            return ret;
        }
    }
}
public class Exe3_1 {
    public static void main (String[] args) {
        System.out.println("2!="+Factorial.calcFactorial(2));
        System.out.println("4!="+Factorial.calcFactorial(4));
        System.out.println("6!="+Factorial.calcFactorial(6));
        System.out.println("10!="+Factorial.calcFactorial(10));
    }
}
```

程序的输出结果为：

```
2!=2
4!=24
6!=720
10!=3628800
```

3-2　编写程序，接收用户从键盘上输入的三个整数 x,y,z，从中选出最大和最小者。

解：新建 Keyboard.java 文件，其内容为：

```
import java.util.Scanner;
import java.io.*;
public class Keyboard {
    static BufferedReader inputStream=new BufferedReader(new InputStreamReader
    (System.in));
    public static int getInteger() {
        try {
            return (Integer.valueOf(inputStream.readLine().trim()).intValue());
        } catch (Exception e) {
            e.printStackTrace();
            return 0;
        }
    }
    public static String getString() {
        try {
            return inputStream.readLine();
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

新建 Exe3_2.java 文件，其内容为：

```
public class Exe3_2 {
    public static void main (String[] args) {
        System.out.print("请输入 x: ");
        int x=Keyboard.getInteger();
        System.out.print("请输入 y: ");
        int y=Keyboard.getInteger();
        System.out.print("请输入 z: ");
        int z=Keyboard.getInteger();
        if (x>y) {
            if (x>z) {
```

```
                System.out.println("最大数为 x: "+x);
                if (y>z) {
                    System.out.println("最小数为 z:"+z);
                }
                else {//y<=z
                    System.out.println("最小数为 y:"+y);
                }
            }
            else {//x<=z
                System.out.println("最大数为 z: "+z);
                System.out.println("最小数为 y:"+y);
            }
        }
        else {//x<=y
            if (x>z) {
                System.out.println("最小数为 z:"+z);
            }
            else {//x<=z
                System.out.println("最小数为 x:"+x);
                if (y>z) {
                    System.out.println("最大数为 y: "+y);
                }
                else {//y<=z
                    System.out.println("最大数为 z: "+z);
                }
            }
        }
    }
}
```

依次输入 1,2,3 时,输出结果为:

```
请输入 x: 1
请输入 y: 2
请输入 z: 3
最小数为 x:1
最大数为 z: 3
```

依次输入 2,1,3 时,输出结果为:

```
请输入 x: 2
请输入 y: 1
请输入 z: 3
最大数为 z: 3
最小数为 y:1
```

依次输入 3,2,1 时,输出结果为:

```
请输入 x: 3
请输入 y: 2
请输入 z: 1
最大数为 x: 3
最小数为 z:1
```

3-3　求出 100 以内的素数,并将这些数在屏幕上 5 个一行地显示出来。

解:新建 Exe3_3.java 文件,其内容为:

```
public class Exe3_3 {
    public static void pritnPrime(int n) {
        boolean[] isPrime=new boolean[n+1];          //表示 0-n 是否是素数
        for (int i=0; i<=n; i++) {
            isPrime[i]=true;
        }
        isPrime[0]=false;                            //0 不是素数
        isPrime[1]=false;                            //1 不是素数
        //依次标记 2,3,4…的倍数不是素数
        for (int i=2; i<=Math.sqrt(n); i++) {
            for (int j=2; i*j<=n; j++) {
                isPrime[i*j]=false;
            }
        }
        int num=0;
        for (int i=0; i<n; i++) {
            if(isPrime[i]) {
                System.out.print(i+"\t");
                num++;
                if(num %5==0)
                System.out.println("");
            }

        }
    }
    public static void main(String[] args) {
       pritnPrime(100);
    }
}
```

程序的输出结果为:

```
2       3       5       7       11
13       17       19       23       29
31       37       41       43       47
53       59       61       67       71
73       79       83       89       97
```

3-4 使用 java.lang.Math 类,生成 100 个 0～99 之间的随机整数,找出它们之中的最大者及最小者,并统计大于 50 的整数个数。

解:新建 Exe3_4.java 文件,其内容为:

```
import java.util.*;
public class Exe3_4 {
    public static void main(String[] args) {
        Random random=new Random();
        int min=100; int max=-1; int count=0;
        for (int i=0; i<100; i++) {
            int a=random.nextInt(100);
            if (a>max)
                max=a;
            if (a<min)
                min=a;
            if (a>50)
                count++;
        }
        System.out.println("最小数为: "+min);
        System.out.println("最大数为: "+max);
        System.out.println("大于 50 的整数个数: "+count);
    }
}
```

程序运行的输出结果为:

```
最小数为: 2
最大数为: 96
大于 50 的整数个数: 52
```

3-5 接收用户从键盘上输入的两个整数,求两个数的最大公约数和最小公倍数,并输出。

解:新建 Exe3_5.java 文件,其内容为:

```
class CalcGCD {
    public static int calcGCG(int a, int b) {
        //使用辗转相除法计算最大公约数
        int max=0; int min=0;
        if (a>b) {
```

```
            max=a;
            min=b;
        }
        else {
            max=b;
            min=a;
        }
        while(max %min != 0) {
            int temp=max %min;
            max=min;
            min=temp;
        }
        return min;
    }
}
class CalcLCM {
    public static int calcLCM(int a, int b) {
        int gcd=CalcGCD.calcGCG(a, b);
        int lcm= (a/gcd) * (b/gcd) * gcd;
        return lcm;
    }
}
public class Exe3_5 {
    public static void main (String[] args) {
        System.out.print("请输入第一个整数: ");
        int a=Keyboard.getInteger();
        System.out.print("请输入第二个整数: ");
        int b=Keyboard.getInteger();
        System.out.print(a+"和"+b+"的最大公约数为: ");
        System.out.println(CalcGCD.calcGCG(a, b));
        System.out.print(a+"和"+b+"的最小公倍数为: ");
        System.out.println(CalcLCM.calcLCM(a, b));
    }
}
```

当输入 6 和 8 时,程序的输出结果如下:

```
请输入第一个整数: 6
请输入第二个整数: 8
6 和 8 的最大公约数为: 2
6 和 8 的最小公倍数为: 24
```

3-6　从键盘上输入一件物品的价格(范围在 0.10～5.00 元),假设用户付了一张 5 元纸币,请列出一种找零的方案,使得纸币及硬币的个数最少。如 3.68 元,应为:两元一张,一元一张,五角一枚,一角一枚,五分一枚,二分一枚,一分一枚。