HARBIN INSTITUTE OF TECHNOLOGY

哈尔滨工业大学

Java 程序设计实验报告

学号：xxxxxxxxxxxx

姓名：xxxxxx

专业：xxxxxxxxxxxx

班级：1xxxxxx

实验二：简单程序设计（控制结构）

一、实验目的

1）掌握自定义 static 方法及其调用
2）掌握 if 和 if-else 和 switch 分支语句
3）掌握 while、do-while 和 for 循环语句
4）掌握 break 和 continue 程序控制语句

二、实验内容

1）在 Eclipse 中输入 **ControlStrcutureTest** 程序，不断修改 a 的值，观察并分析执行结果。如果你觉得该程序有问题，应该如何修改？并将修改后的正确程序，按照 if-else 的语法编制一个功能完全相同的程序。

```
package edu.hitXXX.experiment2;//将 XXX 替换为自己的学号
public class ControlStrcutureTest {
    public static void main(String[] args) {
        switchTest();
    }
    public static void switchTest(){
        char a=2;
        switch(a){
            case 1:System.out.println("I’m Mary");
            case 2:System.out.println("I’m Jack");
            case 3:System.out.println("I’m Trump");
            case 4:System.out.println("I’m Obama");
            default:System.out.println("Who am I?");
        }
    }
}
```

2）在 ControlStrcutureTest 中增加函数 doWhileTest，解释该函数的功能，在 main 函数中调用该函数，并写出该函数的打印结果

```
public static void doWhileTest() {
    int i=1;
    do{
        if(i%2==0)
            System.out.print (i+" ");
        i++;
    }while(i<=20);
}
```

3）在 ControlStrcutureTest 中增加函数 forTest，解释该函数的功能，并写出该函数的打印结果

```
public static void forTest() {
    int sum=0,K=100;
    int i;
    for(i=0;i<=K;i++) sum+=i;
    System.out.printf("1+2+…+%d=%d%n",K,sum);
}
```

4）在 ControlStrcutureTest 中增加一个函数 checkLeapYear，函数参数为整数年份，返回值为 true/false，该函数用于判断某年份是否闰年，在 man 函数中，分别用 220, 618, 960, 1271, 1368, 1644, 1912, 1949, 2000, 2017, 2020 等年份进行测试。闰年判断规则如下：

①能被 4 整除且不能被 100 整除的为闰年。

②能被 400 整除的是闰年。

5）在 ControlStrcutureTest 中增加一个函数 printLeapYears，该函数能够打印出开始年份到结束年份中所有的闰年；在 mian 函数中调用该函数，输出 2001-2050 这 50 年中所有的闰年。

6）为 printLeapYears 函数增加一个参数，使得该函数能够打印出开始年份到结束年份中的第一个闰年；在 mian 函数中调用该函数，输出 2001-2050 这 50 年中的第一个闰年。

7) 在 ControlStrcutureTest 中增加一个如下猜数小游戏的函数。

```
public static guessNumber (){
    System.out.println("给你一个 1 至 100 之间的整数,请猜测这个数");
    int realNumber=(int)(Math.random()*100)+1;
    int yourGuess=0; //你猜的数
    Scanner in=new Scanner(System.in);
    System.out.println("输入您的猜测:");
    _____ //从键盘键入你猜的数
    while (_____){
        if (_____){
            System.out.println("大了,请再猜:");
            _____ //从键盘键入你猜的数
        }else if(_____){
            System.out.println("猜小了,请再猜:");
            _____ //从键盘键入你猜的数
        }
    }
    System.out.println("恭喜！猜对了!");
}
```

你觉得该程序是否可以进行优化？如果可以，怎么优化?

8）在 BMI 程序中，增加一个函数 checkHealth，函数参数为 bmi 值，该函数按下表中 BMI 取值范围判断胖瘦健康状况，该函数的返回值为字符串，返回结果即下表中的第一列中的值，并在 input 函数中调用该函数，并打印输出学生的胖瘦健康状况。

Category	BMI (kg/m^2)	
	from	to
Underweight		18.5
Normal Range	18.5	23
Overweight—At Risk	23	25
Overweight—Moderately Obese	25	30
Overweight—Severely Obese	30	

三、实验步骤

1）在 Eclipse 中输入 **ControlStrcutureTest** 程序，不断修改 a 的值，观察并分析执行结果。如果你觉得该程序有问题，应该如何修改？并将修改后的正确程序，按照 if-else 的语法编制一个功能完全相同的程序。

Step1:修改程序如下，并运行观察结果

ControlStrcutureTest.java

```
1  package edu.hit1162800204.experiment2;
2
3  public class ControlStrcutureTest {
4
5      public static void main(String[] args){
6
7
8          for(int i=0;i<5;i++)switchTest(i);
9
10     }
11
12     public static void switchTest(int a){
13         switch(a)   {
14             case 1:System.out.println("I'm Mary");break;
15             case 2:System.out.println("I'm Jack");break;
16             case 3:System.out.println("I'm Trump");break;
17             case 4:System.out.println("I'm Obama");break;
18             default:System.out.println("Who am I?");break;
19         }
20     }
21
22 }
23
```

运行结果如下：

```
<terminated> BMI [Java A
Who am I?
I'm Mary
I'm Jack
I'm Trump
I'm Obama
```

Step2：按照 if-else 的语法编写如下：

```
public static void ifTest(int a) {
    if(a==1)
        System.out.println("I'm Mary");
    else if (a==2)
        System.out.println("I'm Jack");
    else if (a==3)
        System.out.println("I'm Trump");
    else if (a==4)
        System.out.println("I'm Obama");
    else
        System.out.println("Who am I?");
    }
```

运行结果如下：

```
<terminated> BMI [Java A
Who am I?
I'm Mary
I'm Jack
I'm Trump
I'm Obama
```

2）在 ControlStrcutureTest 中增加函数 doWhileTest，解释该函数的功能，在 main 函数中调用该函数，并写出该函数的打印结果

Step1:添加函数如下，功能为输出 1-20 中所有偶数

```
public static void doWhileTest() {
    int i=1;
    do{
        if(i%2==0)
            System.out.print (i+" ");
        i++;
    }while(i<=20);
}
```

Step2：在main中调用并运行观察结果

```
public static void main(String[] args){
    int i=0;
    doWhileTest();
    //for(i=0;i<5;i++)switchTest(i);
    //for(i=0;i<5;i++)ifTest(i);

}
```

结果如下：

```
Problems  @ Javadoc  Declaration
<terminated> BMI [Java Application] D:\Pr
2 4 6 8 10 12 14 16 18 20
```

3）在 ControlStrcutureTest 中增加函数 forTest，解释该函数的功能，并写出该函数的打印结果

Step1：添加函数如下，功能为输出 1-100 中所有数的和

```
public static void forTest() {
    int sum=0,K=100;
    int i;
    for(i=0;i<=K;i++) sum+=i;
    System.out.printf("1+2+…+%d=%d%n",K,sum);
}
```

Step2：在main中调用并运行观察结果

```
public static void main(String[] args){
    int i=0;
    forTest();
    //doWhileTest();
    //for(i=0;i<5;i++)switchTest(i);
    //for(i=0;i<5;i++)ifTest(i);

}
```

运行结果如下：

```
Problems  @ Javadoc
<terminated> BMI [Java Ap
1+2+…+100=5050
```

4）在 ControlStrcutureTest 中增加一个函数 checkLeapYear，函数参数为整数年份，返回值为 true/false，该函数用于判断某年份是否闰年，在 man 函数中，分别用 220, 618, 960, 1271, 1368, 1644, 1912, 1949, 2000, 2017, 2020 等年份进行测试。闰年判断规则如下：

①能被 4 整除且不能被 100 整除的为闰年。
②能被 400 整除的是闰年。

Step1：添加函数如下

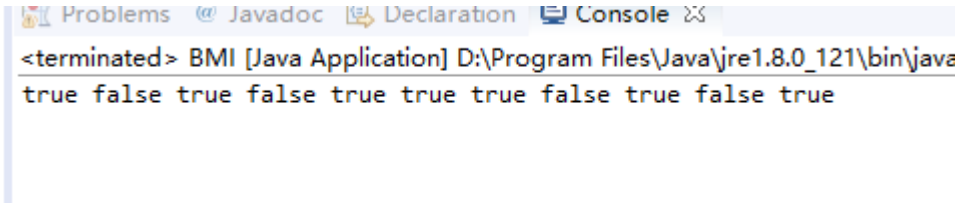
```
public static boolean checkLeapYear(int year){
    if(year%4==0 && year%100!=0||year%400==0)return true;
    else return false;
}
```

Step2：在main中调用并运行观察结果

```
public static void main(String[] args){
    int i=0;
    int years[]={220, 618, 960, 1271, 1368, 1644, 1912, 1949, 2000, 2017, 2020};
    for(i=0;i<11;i++)System.out.printf("%b ",checkLeapYear( years[i]));
    //forTest();
    //doWhileTest();
    //for(i=0;i<5;i++)switchTest(i);
    //for(i=0;i<5;i++)ifTest(i);

}
```

运行结果如下：



5）在 ControlStrcutureTest 中增加一个函数 printLeapYears，该函数能够打印出开始年份到结束年份中所有的闰年；在 mian 函数中调用该函数，输出 2001-2050 这 50 年中所有的闰年。

Step1：添加函数如下

```
public static void printLeapYears(){
    for(int year=2001;year<2051;year++)
        if(year%4==0 && year%100!=0||year%400==0)
            System.out.printf("%d ",year);

}
```

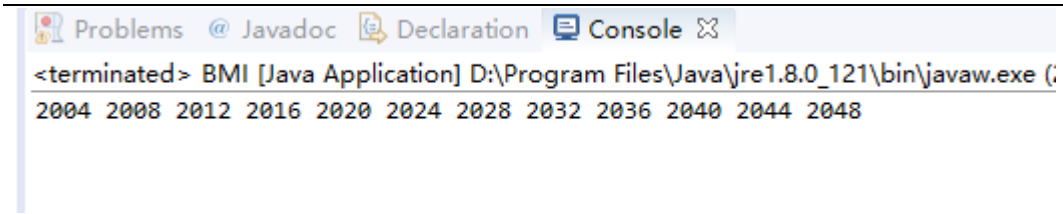
Step2：在main中调用并运行观察结果

```
public static void main(String[] args){

    printLeapYears();
    //int i=0;
    //int years[]={220, 618, 960, 1271, 1368,
    //for(i=0;i<11;i++)System.out.printf("%b
    //forTest();
    //doWhileTest();
    //for(i=0;i<5;i++)switchTest(i);
    //for(i=0;i<5;i++)ifTest(i);

}
```

运行结果如下：



6）为 printLeapYears 函数增加一个参数，使得该函数能够打印出开始年份到结束年份中的第一个闰年；在 mian 函数中调用该函数，输出 2001-2050 这 50 年中的第一个闰年。

Step1：添加函数如下

```
public static void printFistLeapYears(int begin,int end){
    for(int year=begin;year<=end;year++)
        if(year%4==0 && year%100!=0||year%400==0){
            System.out.printf("%d ",year);
            return;
        }

}
```

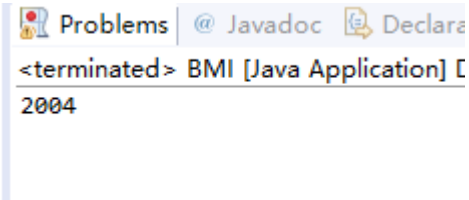
Step2：在main中调用并运行观察结果

```
  public static void main(String[] args){

      printFistLeapYears(2001,2050);
      //printLeapYears();
      //int i=0;
      //int years[]={220, 618, 960, 1271, 136
      //for(i=0;i<11;i++)System.out.printf("%
      //forTest();
      //doWhileTest();
      //for(i=0;i<5;i++)switchTest(i);
      //for(i=0;i<5;i++)ifTest(i);

  }
```

运行结果如下：



7）在 ControlStrcutureTest 中增加一个如下猜数小游戏的函数。

你觉得该程序是否可以进行优化？如果可以，怎么优化?

Step1：添加函数如下

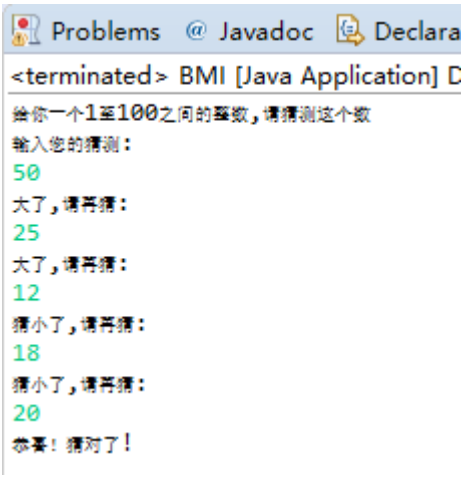
```
public static void guessNumber (){
    System.out.println("给你一个1至100之间的整数,请猜测这个数");
    int realNumber=(int)(Math.random()*100)+1;
    int yourGuess=0; //你猜的数
    Scanner in=new Scanner(System.in);
    System.out.println("输入您的猜测:");
    yourGuess=in.nextInt(); //从键盘键入你猜的数
    while (yourGuess!=realNumber){
        if (yourGuess>realNumber){
            System.out.println("大了,请再猜:");
            yourGuess=in.nextInt();//从键盘键入你猜的数
        }else if(yourGuess<realNumber){
            System.out.println("猜小了,请再猜:");
            yourGuess=in.nextInt(); //从键盘键入你猜的数
        }
    }
        System.out.println("恭喜!猜对了!");
}
```

Step2：运行并观察结果

改程序可改进为 do-while 循环如下：

```
public static void guessNumber(){
    System.out.println("给你一个1至100之间的整数,请猜测这个数");
    int realNumber=(int)(Math.random()*100)+1;
    int yourGuess=0; //你猜的数
    Scanner in=new Scanner(System.in);
    System.out.println("输入您的猜测:");
    do{
        yourGuess=in.nextInt(); //从键盘键入你猜的数
        if (yourGuess>realNumber){
            System.out.println("大了,请再猜:");
        }else if(yourGuess<realNumber){
            System.out.println("猜小了,请再猜:");
        }
    }while (yourGuess!=realNumber);
        System.out.println("恭喜!猜对了!");
}
```

运行结果如下：



8）在 BMI 程序中，增加一个函数 checkHealth，函数参数为 bmi 值，该函数按下表中 BMI 取值范围判断胖瘦健康状况，该函数的返回值为字符串，返回结果即下表中的第一列中的值，并在 input 函数中调用该函数，并打印输出学生的胖瘦健康状况。

Category	BMI (kg/m^2)	
	from	to
Underweight		18.5
Normal Range	18.5	23
Overweight—At Risk	23	25
Overweight—Moderately Obese	25	30
Overweight—Severely Obese	30	

Step1：添加 checkHealth 函数如下：

```
public static String checkHealth(float bmi){
    String str[]={"Underweight","Normal Range",
            "Overweight-At Risk","Overweight-Moderately Obese",
            "Overweight-Severely Obese"};
    if(bmi<18.5)return str[0];
    else if(bmi<23)return str[1];
    else if(bmi<25)return str[2];
    else if(bmi<30)return str[3];
    else return str[4];
    
}
```

Step1：添加 input 函数如下：

```
public static void Input(){

    Scanner input=new Scanner(System.in);
    float bmi=input.nextFloat();
    System.out.println(checkHealth(bmi));
}
```

运行结果如下：