



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 ControlStrcutreTest 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 ControlStrcutreTest 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 ControlStrcutreTest 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         for(int i=0;i<5;i++)switchTest(i);
8     }
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 /
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 /  
Who am I?  
I'm Mary  
I'm Jack  
I'm Trump  
I'm Obama
```

2) 在 ControlStrcutreTest 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 ControlStrcutreTest 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

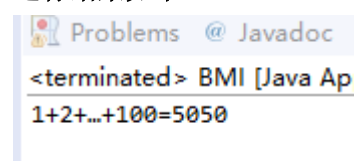
Step1: 添加函数如下, 功能为输出 1-100 中所有数的和

```
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 ControlStrcutreTest 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;  
}
```

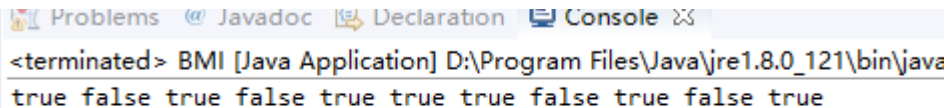
Step2: 在main中调用并运行观察结果

```

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：



```

<terminated> BMI [Java Application] D:\Program Files\Java\jre1.8.0_121\bin\java
true false true false true true true false true false true

```

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 main 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, 1360, 1672, 1984, 2296, 2608, 2920, 3232, 3544, 3856, 4168, 4480, 4792, 5104, 5416, 5728, 6040, 6352, 6664, 6976, 7288, 7600, 7912, 8224, 8536, 8848, 9160, 9472, 9784, 10096, 10408, 10720, 11032, 11344, 11656, 11968, 12280, 12592, 12904, 13216, 13528, 13840, 14152, 14464, 14776, 15088, 15400, 15712, 16024, 16336, 16648, 16960, 17272, 17584, 17896, 18208, 18520, 18832, 19144, 19456, 19768, 20080, 20392, 20704, 21016, 21328, 21640, 21952, 22264, 22576, 22888, 23200, 23512, 23824, 24136, 24448, 24760, 25072, 25384, 25696, 26008, 26320, 26632, 26944, 27256, 27568, 27880, 28192, 28504, 28816, 29128, 29440, 29752, 30064, 30376, 30688, 31000, 31312, 31624, 31936, 32248, 32560, 32872, 33184, 33496, 33808, 34120, 34432, 34744, 35056, 35368, 35680, 35992, 36304, 36616, 36928, 37240, 37552, 37864, 38176, 38488, 38800, 39112, 39424, 39736, 40048, 40360, 40672, 40984, 41296, 41608, 41920, 42232, 42544, 42856, 43168, 43480, 43792, 44104, 44416, 44728, 45040, 45352, 45664, 45976, 46288, 46600, 46912, 47224, 47536, 47848, 48160, 48472, 48784, 49096, 49408, 49720, 50032, 50344, 50656, 50968, 51280, 51592, 51904, 52216, 52528, 52840, 53152, 53464, 53776, 54088, 54400, 54712, 55024, 55336, 55648, 55960, 56272, 56584, 56896, 57208, 57520, 57832, 58144, 58456, 58768, 59080, 59392, 59704, 60016, 60328, 60640, 60952, 61264, 61576, 61888, 62200, 62512, 62824, 63136, 63448, 63760, 64072, 64384, 64696, 65008, 65320, 65632, 65944, 66256, 66568, 66880, 67192, 67504, 67816, 68128, 68440, 68752, 69064, 69376, 69688, 69992, 70304, 70616, 70928, 71240, 71552, 71864, 72176, 72488, 72800, 73112, 73424, 73736, 74048, 74360, 74672, 74984, 75296, 75608, 75920, 76232, 76544, 76856, 77168, 77480, 77792, 78104, 78416, 78728, 79040, 79352, 79664, 79976, 80288, 80600, 80912, 81224, 81536, 81848, 82160, 82472, 82784, 83096, 83408, 83720, 84032, 84344, 84656, 84968, 85280, 85592, 85904, 86216, 86528, 86840, 87152, 87464, 87776, 88088, 88400, 88712, 89024, 89336, 89648, 89960, 90272, 90584, 90896, 91208, 91520, 91832, 92144, 92456, 92768, 93080, 93392, 93704, 94016, 94328, 94640, 94952, 95264, 95576, 95888, 96200, 96512, 96824, 97136, 97448, 97760, 98072, 98384, 98696, 99008, 99320, 99632, 99944, 100256, 100568, 100880, 101192, 101504, 101816, 102128, 102440, 102752, 103064, 103376, 103688, 104000, 104312, 104624, 104936, 105248, 105560, 105872, 106184, 106496, 106808, 107120, 107432, 107744, 108056, 108368, 108680, 108992, 109304, 109616, 109928, 110240, 110552, 110864, 111176, 111488, 111800, 112112, 112424, 112736, 113048, 113360, 113672, 113984, 114296, 114608, 114920, 115232, 115544, 115856, 116168, 116480, 116792, 117104, 117416, 117728, 118040, 118352, 118664, 118976, 119288, 119600, 119912, 120224, 120536, 120848, 121160, 121472, 121784, 122096, 122408, 122720, 123032, 123344, 123656, 123968, 124280, 124592, 124904, 125216, 125528, 125840, 126152, 126464, 126776, 127088, 127400, 127712, 128024, 128336, 128648, 128960, 129272, 129584, 129896, 130208, 130520, 130832, 131144, 131456, 131768, 132080, 132392, 132704, 133016, 133328, 133640, 133952, 134264, 134576, 134888, 135200, 135512, 135824, 136136, 136448, 136760, 137072, 137384, 137696, 138008, 138320, 138632, 138944, 139256, 139568, 139880, 140192, 140504, 140816, 141128, 141440, 141752, 142064, 142376, 142688, 143000, 143312, 143624, 143936, 144248, 144560, 144872, 145184, 145496, 145808, 146120, 146432, 146744, 147056, 147368, 147680, 147992, 148304, 148616, 148928, 149240, 149552, 149864, 150176, 150488, 150800, 151112, 151424, 151736, 152048, 152360, 152672, 152984, 153296, 153608, 153920, 154232, 154544, 154856, 155168, 155480, 155792, 156104, 156416, 156728, 157040, 157352, 157664, 157976, 158288, 158600, 158912, 159224, 159536, 159848, 160160, 160472, 160784, 161096, 161408, 161720, 162032, 162344, 162656, 162968, 163280, 163592, 163904, 164216, 164528, 164840, 165152, 165464, 165776, 166088, 166400, 166712, 167024, 167336, 167648, 167960, 168272, 168584, 168896, 169208, 169520, 169832, 170144, 170456, 170768, 171080, 171392, 171704, 172016, 172328, 172640, 172952, 173264, 173576, 173888, 174200, 174512, 174824, 175136, 175448, 175760, 176072, 176384, 176696, 177008, 177320, 177632, 177944, 178256, 178568, 178880, 179192, 179504, 179816, 180128, 180440, 180752, 181064, 181376, 181688, 182000, 182312, 182624, 182936, 183248, 183560, 183872, 184184, 184496, 184808, 185120, 185432, 185744, 186056, 186368, 186680, 186992, 187304, 187616, 187928, 188240, 188552, 188864, 189176, 189488, 189800, 190112, 190424, 190736, 191048, 191360, 191672, 191984, 192296, 192608, 192920, 193232, 193544, 193856, 194168, 194480, 194792, 195104, 195416, 195728, 196040, 196352, 196664, 196976, 197288, 197600, 197912, 198224, 198536, 198848, 199160, 199472, 199784, 200096, 200408, 200720, 201032, 201344, 201656, 201968, 202280, 202592, 202904, 203216, 203528, 203840, 204152, 204464, 204776, 205088, 205400, 205712, 206024, 206336, 206648, 206960, 207272, 207584, 207896, 208208, 208520, 208832, 209144, 209456, 209768, 210080, 210392, 210704, 211016, 211328, 211640, 211952, 212264, 212576, 212888, 213200, 213512, 213824, 214136, 214448, 214760, 215072, 215384, 215696, 216008, 216320, 216632, 216944, 217256, 217568, 217880, 218192, 218504, 218816, 219128, 219440, 219752, 220064, 220376, 220688, 221000, 221312, 221624, 221936, 222248, 222560, 222872, 223184, 223496, 223808, 224120, 224432, 224744, 225056, 225368, 225680, 225992, 226304, 226616, 226928, 227240, 227552, 227864, 228176, 228488, 228800, 229112, 229424, 229736, 230048, 230360, 230672, 230984, 231296, 231608, 231920, 232232, 232544, 232856, 233168, 233480, 233792, 234104, 234416, 234728, 235040, 235352, 235664, 235976, 236288, 236600, 236912, 237224, 237536, 237848, 238160, 238472, 238784, 239096, 239408, 239720, 240032, 240344, 240656, 240968, 241280, 241592, 241904, 242216, 242528, 242840, 243152, 243464, 243776, 244088, 244400, 244712, 245024, 245336, 245648, 245960, 246272, 246584, 246896, 247208, 247520, 247832, 248144, 248456, 248768, 249080, 249392, 249704, 250016, 250328, 250640, 250952, 251264, 251576, 251888, 252200, 252512, 252824, 253136, 253448, 253760, 254072, 254384, 254696, 255008, 255320, 255632, 255944, 256256, 256568, 256880, 257192, 257504, 257816, 258128, 258440, 258752, 259064, 259376, 259688, 259992, 260304, 260616, 260928, 261240, 261552, 261864, 262176, 262488, 262800, 263112, 263424, 263736, 264048, 264360, 264672, 264984, 265296, 265608, 265920, 266232, 266544, 266856, 267168, 267480, 267792, 268104, 268416, 268728, 269040, 269352, 269664, 269976, 270288, 270600, 270912, 271224, 271536, 271848, 272160, 272472, 272784, 273096, 273408, 273720, 274032, 274344, 274656, 274968, 275280, 275592, 275904, 276216, 276528, 276840, 277152, 277464, 277776, 278088, 278400, 278712, 279024, 279336, 279648, 279960, 280272, 280584, 280896, 281208, 281520, 281832, 282144, 282456, 282768, 283080, 283392, 283704, 284016, 284328, 284640, 284952, 285264, 285576, 285888, 286200, 286512, 286824, 287136, 287448, 287760, 288072, 288384, 288696, 289008, 289320, 289632, 289944, 290256, 290568, 290880, 291192, 291504, 291816, 292128, 292440, 292752, 293064, 293376, 293688, 294000, 294312, 294624, 294936, 295248, 295560, 295872, 296184, 296496, 296808, 297120, 297432, 297744, 298056, 298368, 298680, 298992, 299304, 299616, 299928, 300240, 300552, 300864, 301176, 301488, 301800, 302112, 302424, 302736, 303048, 303360, 303672, 303984, 304296, 304608, 304920, 305232, 305544, 305856, 306168, 306480, 306792, 307104, 307416, 307728, 308040, 308352, 308664, 308976, 309288, 309600, 309912, 310224, 310536, 310848, 311160, 311472, 311784, 312096, 312408, 312720, 313032, 313344, 313656, 313968, 314280, 314592, 314904, 315216, 315528, 315840, 316152, 316464, 316776, 317088, 317400, 317712, 318024, 318336, 318648, 318960, 319272, 319584, 319896, 320208, 320520, 320832, 321144, 321456, 321768, 322080, 322392, 322704, 323016, 323328, 323640, 323952, 324264, 324576, 324888, 325200, 325512, 325824, 326136, 326448, 326760, 327072, 327384, 327696, 328008, 328320, 328632, 328944, 329256, 329568, 329880, 330192, 330504, 330816, 331128, 331440, 331752, 332064, 332376, 332688, 333000, 333312, 333624, 333936, 334248, 334560, 334872, 335184, 335496, 335808, 336120, 336432, 336744, 337056, 337368, 337680, 337992, 338304, 338616, 338928, 339240, 339552, 339864, 340176, 340488, 340800, 341112, 341424, 341736, 342048, 342360, 342672, 342984, 343296, 343608, 343920, 344232, 344544, 344856, 345168, 345480, 345792, 346104, 346416, 346728, 347040, 347352, 347664, 347976, 348288, 348600, 348912, 349224, 349536, 349848, 350160, 350472, 350784, 351096, 351408, 351720, 352032, 352344, 352656, 352968, 353280, 353592, 353904, 354216, 354528, 354840, 355152, 355464, 355776, 356088, 356400, 356712, 357024, 357336, 357648, 357960, 358272, 358584, 358896, 359208, 359520, 359832, 360144, 360456, 360768, 361080, 361392, 361704, 362016, 362328, 362640, 362952, 363264, 363576, 363888, 364200, 364512, 364824, 365136, 365448, 365760, 366072, 366384, 366696, 367008, 367320, 367632, 367944, 368256, 368568, 368880, 369192, 369504, 369816, 370128, 370440, 370752, 371064, 371376, 371688, 372000, 372312, 372624, 372936, 373248, 373560, 373872, 374184, 374496, 374808, 375120, 375432, 375744, 376056, 376368, 376680, 376992, 377304, 377616, 377928, 378240, 378552, 378864, 379176, 379488, 379800, 380112, 380424, 380736, 381048, 381360, 381672, 381984, 382296, 382608, 382920, 383232, 383544, 383856, 384168, 384480, 384792, 385104, 385416, 385728, 386040, 386352, 386664, 386976, 387288, 387600, 387912, 388224, 388536, 388848, 389160, 389472, 389784, 390096, 390408, 390720, 391032, 391344, 391656, 391968, 392280, 392592, 392904, 393216, 393528, 393840, 394152, 394464, 394776, 395088, 395400, 395712, 396024, 396336, 396648, 396960, 397272, 397584, 397896, 398208, 398520, 398832, 399144, 399456, 399768, 400080, 400392, 400704, 401016, 401328, 401640, 401952, 402264, 402576, 402888, 403200, 403512, 403824, 404136, 404448, 404760, 405072, 405384, 405696, 406008, 406320, 406632, 406944, 407256, 407568, 407880, 408192, 408504, 408816, 409128, 409440, 409752, 410064, 410376, 410688, 411000, 411312, 411624, 411936, 412248, 412560, 412872, 413184, 413496, 413808, 414120, 414432, 414744, 415056, 415368, 415680, 415992, 416304, 416616, 416928, 417240, 417552, 417864, 418176, 418488, 418800, 419112, 419424, 419736, 420048, 420360, 420672, 420984, 421296, 421608, 421920, 422232, 422544, 422856, 423168, 423480, 423792, 424104, 424416, 424728, 425040, 425352, 425664, 425976, 426288, 426600, 426912, 427224, 427536, 427848, 428160, 428472, 428784, 429096, 429408, 429720, 430032, 430344, 430656, 430968, 431280, 431592, 431904, 432216, 432528, 432840, 433152, 433464, 433776, 434088, 434400, 434712, 435024, 435336, 435648, 435960, 436272, 436584, 436896, 437208, 437520, 437832, 438144, 438456, 438768, 439080, 439392, 439704, 440016, 440328, 440640, 440952, 441264, 441576, 441888, 442200, 442512, 442824, 443136, 443448, 443760, 444072, 444384, 444696, 445008, 445320, 445632, 445944, 446256, 446568, 446880, 447192, 447504, 447816, 448128, 448440, 448752, 449064, 449376, 449688, 449992, 450304, 450616, 450928, 451240, 451552, 451864, 452176, 452488, 452800, 453112, 453424, 453736, 454048, 454360, 454672, 454984, 455296, 455608, 455920, 456232, 456544, 456856, 457168, 457480, 457792, 458104, 458416, 458728, 459040, 459352, 459664, 459976, 460288, 460600, 460912, 461224, 461536, 461848, 462160, 462472, 462784, 463096, 463408, 463720, 464032, 464344, 464656, 464968, 465280, 465592, 465904, 466216, 466528, 466840, 467152, 467464, 467776, 468088, 468400, 468712, 469024, 469336, 469648, 469960, 470272, 470584, 470896, 471208, 471520, 471832, 472144, 47245
```



```

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(); //从键盘输入你猜的数
    while (yourGuess!=realNumber){
        if (yourGuess>realNumber){
            System.out.println("大了, 请再猜:");
            yourGuess=in.nextInt(); //从键盘输入你猜的数
        }else if(yourGuess<realNumber){
            System.out.println("猜小了, 请再猜:");
            yourGuess=in.nextInt(); //从键盘输入你猜的数
        }
    }
    System.out.println("恭喜! 猜对了!");
}

```

Step2: 运行并观察结果



```

Problems @ Javadoc Declarat
<terminated> BMI [Java Application] D
给你一个1至100之间的整数, 请猜测这个数
输入您的猜测:
50
大了, 请再猜:
25
大了, 请再猜:
12
恭喜! 猜对了!

```

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(); //从键盘输入你猜的数
        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:

```

Problems @ Javadoc Declara
<terminated> BMI [Java Application] C
给你一个1至100之间的整数,请猜测这个数
输入您的猜测:
50
大了,请再猜:
25
大了,请再猜:
12
猜小了,请再猜:
18
猜小了,请再猜:
20
恭喜!猜对了!

```

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：

