

上机实践 3

分支与循环语句

实验 1 回 文 数

1. 相关知识点

1) 分支语句

if-else 语句是 Java 中的一条语句，由关键字 if、else 和两个复合语句（分别称为 if 分支操作和 else 分支操作）按一定格式构成，if-else 语句的格式如下：

```
if(表达式) {  
    若干语句 (if分支操作部分)  
}  
else {  
    若干语句 (else分支操作部分)  
}
```

一条 if-else 语句的作用是根据一个条件选择两个分支操作中的一个，执行法则是：if 后面()内的表达式的值为 true，则执行紧跟着的复合语句，即执行 if 分支操作；如果表达式的值为 false，则执行 else 后面的复合语句，即执行 else 分支操作。

if-else if-else 语句称为多条件分支语句，其作用是根据多条件来选择某一操作。语句的格式如下：

```
if(表达式1) {  
    若干语句  
}  
else if(表达式2) {  
    若干语句  
}  
:  
else {  
    若干语句  
}
```

2) 将字符串转化为数值

执行 “int m=Integer.parseInt("6789");” 可以将数字型字符串，如 “6789”、“123”，转化为 int 型数据。

2. 实验目的

本实验的目的是让学生掌握使用 if-else if-else 多分支语句。

3. 实验要求

编写一个 Java 应用程序程序。用户从键盘输入一个 1~9999 之间的数，程序将判断这个数是几位数，并判断这个数是否是回文数。回文数是指将该数含有的数字逆序排列后得到的数和原数相同，例如 12121、3223 都是回文数。

4. 程序效果示例

程序运行效果如图 3.1 所示。

5. 程序模板

请按模板要求，将【代码】替换为 Java 程序

代码。

```
输入一个1至99999之间的数
9889
9889是4位数
9889是回文数
```

图 3.1 判断回文数

Number.java

```
import java.util.Scanner;

public class Number {
    public static void main(String args[]) {
        int number=0,d5,d4,d3,d2,d1;
        Scanner reader = new Scanner(System.in);
        System.out.println("输入一个1至99999之间的数");
        number = reader.nextInt();
        if(【代码1】) //判断number在1~99999之间的条件
        {
            【代码2】 //计算number的最高位(万位)d5
            【代码3】 //计算number的千位d4
            【代码4】 //计算number的百位d3
            d2=number%100/10;
            d1=number%10;
            if(【代码5】) //判断number是5位数的条件
            {
                System.out.println(number+"是5位数");
                if(【代码6】) //判断number是回文数的条件
                {
                    System.out.println(number+"是回文数");
                }
                else
                {
                    System.out.println(number+"不是回文数");
                }
            }
            else if(【代码7】) //判断number是4位数的条件
            {
                System.out.println(number+"是4位数");
                if(【代码8】) //判断number是回文数的条件码
                {
                    System.out.println(number+"是回文数");
                }
            }
            else
            {
                System.out.println(number+"不是回文数");
            }
        }
        else if(【代码9】) //判断number是3位数的条件
        {
            System.out.println(number+"是3位数");
            if(【代码10】) //判断number是回文数的条件
            {
                System.out.println(number+"是回文数");
            }
        }
    }
}
```



```
    }
    else
    { System.out.println(number+"不是回文数");
    }
}
else if (d2!=0)
{ System.out.println(number+"是2位数");
  if (d1==d2)
  { System.out.println(number+"是回文数");
  }
  else
  { System.out.println(number+"不是回文数");
  }
}
else if (d1!=0)
{ System.out.println(number+"是1位数");
  System.out.println(number+"是回文数");
}
}
else
{ System.out.printf("\n%d不在1至99999之间", number);
}
}
```

6. 实验指导与检查

- ◇ 两个 int 型数据做除法运算时, 运算的结果是 int 型, 因此, 6234/1000 的结果刚好是 6234 的最高位上的数字 6。
- ◇ 为了计算出 56321 中千位上的数字 6, 首先计算 56321%10000 得到 6321, 然后 6321/1000 的结果是 6。

7. 实验后的练习

- (1) 程序运行时, 用户从键盘输入 2332, 程序提示怎样的信息。
- (2) 程序运行时, 用户从键盘输入 654321, 程序提示怎样的信息。
- (3) 程序运行时, 用户从键盘输入 33321, 程序提示怎样的信息。

8. 填写实验报告

实验编号: 301 学生姓名: 实验时间: 教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后的练习效果评价	A	B	C	D	E
练习 (1) 完成情况					
练习 (2) 完成情况					
练习 (3) 完成情况					
总评					

实验2 猜数字游戏

1. 相关知识点

循环是控制结构语句中的最重要的语句之一，循环语句是根据条件反复执行同一代码块。循环语句有下列三种：

1) while 循环

while 语句的一般格式：

```
while(表达式){
    若干语句
}
```

while 语句的执行规则如下：

- (1) 计算表达式的值，如果该值是 true 时，就进行 (2)，否则进行 (3)。
- (2) 执行循环体，再进行 (1)。
- (3) 结束 while 语句的执行。

2) for 循环

for 语句的一般格式：

```
for (表达式1;表达式2;表达式3) {
    若干语句
}
```

for 语句的执行规则如下：

- (1) 计算“表达式1”，完成必要的初始化工作。
 - (2) 判断“表达式2”的值，若“表达式2”的值为 true，则进行 (3)，否则进行 (4)。
 - (3) 执行循环体，然后计算“表达式3”，以便改变循环条件，进行 (2)。
 - (4) 结束 for 语句的执行。
- 3) 将字符串转化为数值

执行“int m=Integer.parseInt("6789");”可以将数字型字符串，如“6789”、“123”转化为 int 型数据。

4) Scanner 类创建一个对象

```
Scanner reader=new Scanner(System.in);
```

reader 对象调用方法 nextInt()方法读取用户输入的整数。

2. 实验目的

本实验的目的是让学生使用 if-else 分支和 while 循环语句解决问题。

3. 实验要求

编写一个 Java 应用程序，实现下列功能：

- 程序随机分配给客户一个 1~100 之间的整数。
- 用户输入自己的猜测。
- 程序返回提示信息，提示信息分别是“猜大了”、“猜小了”或“猜对了”。

- 用户可根据提示信息再次输入猜测, 直到提示信息是“猜对了”。

4. 程序效果示例

程序运行效果如图 3.2 所示。

5. 程序模板

请按模板要求, 将【代码】替换为 Java 程序代码。

GuessNumber.java

```
import java.util.Scanner;
import java.util.Random;
public class GuessNumber {
    public static void main (String args[]) {
        Scanner reader = new Scanner(System.in);
        Random random = new Random();
        System.out.println("给你一个1至100之间的整数, 请猜测这个数");
        int realNumber = random.nextInt(100)+1; //random.nextInt(100)是[0,100)
                                                //中的随机整数

        int yourGuess = 0;
        System.out.print("输入您的猜测:");
        yourGuess = reader.nextInt();
        while(【代码1】) //循环条件
        {
            if(【代码2】)    //猜大了的条件代码
            {
                System.out.print("猜大了, 再输入你的猜测:");
                yourGuess = reader.nextInt();
            }
            else if(【代码3】) //猜小了的条件代码
            {
                System.out.print("猜小了, 再输入你的猜测:");
                yourGuess = reader.nextInt();
            }
        }
        System.out.println("猜对了!");
    }
}
```

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
猜大了, 再输入你的猜测: 25
猜小了, 再输入你的猜测: 36
猜大了, 再输入你的猜测: 30
猜大了, 再输入你的猜测: 27
猜大了, 再输入你的猜测: 26
猜对了!

图 3.2 猜数字

6. 实验指导

- ✧ 人们经常使用 while 循环“强迫”程序重复执行一段代码, 【代码 1】必须是值为 boolean 型数据的表达式, 只要【代码 1】的值为 true 就是让用户继续输入猜测。
- ✧ 只要用户的输入能使得循环语句结束, 就说明用户已经猜对了。

7. 实验后的练习

- (1) 用“yourGuess>realNumber”替换【代码 1】可以吗?
- (2) 语句“System.out.println("猜对了!");”为何要放在 while 循环语句之后? 放在 while

语句的循环体中合理吗?

8. 填写实验报告

实验编号: 302 学生姓名: 实验时间: 教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后的练习效果评价	A	B	C	D	E
练习(1)完成情况					
练习(2)完成情况					
总评					

实验答案

实验 1:

【代码 1】`number<=99999&&number>=1`

【代码 2】`d5 = number/10000;`

【代码 3】`d4 = number%10000/1000;`

【代码 4】`d3 = number%1000/100;`

【代码 5】`d5!=0`

【代码 6】`d1==d5&&d2==d4`

【代码 7】`d4!=0`

【代码 8】`d1==d4&&d2==d3`

【代码 9】`d3!=0`

【代码 10】`d3==d1`

实验 2:

【代码 1】`yourGuess!=realNumber`

【代码 2】`yourGuess>realNumber`

【代码 3】`yourGuess<realNumber`

自 测 题

1. 下列哪个表达式的值是 false?

(A) `0.6F==0.6`

(B) `12L==12`

(C) `(int)56.98+1==57`

(D) `5<4||10<19`

2. 请说出下列程序的输出结果。

```
public class LianXi3 {  
    public static void main(String args[]) {  
        for(int i=1;i<=4;i++) {
```

```

        switch(i)
        {
            case 2:
                System.out.print("B");
            case 3:
                System.out.print("C");
                break;
            case 1:
                System.out.print("A");
            case 4:
                System.out.print("D");
                break;
        }
    }
}

```

3. 请说出下列程序的输出结果。

```

public class LianXi1 {
    public static void main(String args[]) {
        char a[]={'A','B','C','D','E'};
        for(int i=0;i<=a.length/2;i++){
            char c=a[i];
            a[i]=a[a.length-(i+1)];
            a[a.length-(i+1)]=c;
        }
        for(int i=0;i<a.length;i++) {
            System.out.print(a[i]);
        }
    }
}

```

4. 下列程序用折半法查找一个整数是否是数组中的某个元素, 请说出程序中的【代码】应当是下列中的哪一个。

- (A) $n==a[\text{middle}]$
- (B) $n!=a[\text{middle}]$
- (C) $n!=a[\text{start}]$
- (D) $n!=a[\text{end}]$

```

public class LianXi2 {
    public static void main(String args[]) {
        int start,end,middle,n=-2;
        int a[]={-2,1,4,5,8,12,17,23,45,56,90,100};
        start=0;
        end=a.length;

```



```

middle=(start+end)/2;
int count=0;
while(【代码】) {
    if(n>a[middle])
        start=middle;
    else if(n<a[middle])
        end=middle;
    middle=(start+end)/2;
    count++;
    if(count>a.length/2)
        break;
}
if(count>a.length/2)
    System.out.println(":"+n+"不在数组中");
else
    System.out.println(":"+n+"是数组中的第"+middle+"个元素");
}
}

```

答案:

1. A
2. ADBCCD
3. EDCBA
4. B

Vehicle
speed:double
power:int
speedUp(p:int):void
speedDown(m:int):void
getSpeed():double
setPower(m:int):void
getPower():int