

# 语言基础第四天：

## 笔记：

### 1. Scanner接收用户输入的数据：（分三步）

```
package day04;

import java.util.Scanner; //1.导入扫描仪

//Scanner结构的演示
public class ScannerDemo {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in); //2.新建一个扫描仪scan
        System.out.println("请输入年龄:");
        int age = scan.nextInt(); //3.扫描一个整数并赋值给age
        System.out.println("请输入商品价格:");
        double price = scan.nextDouble(); //3.扫描一个小数并赋值给price
        System.out.println("年龄为:" + age + ", 商品价格为:" + price);
    }
}
```

### 2. 分支结构：

- o if...else if: 多条路

#### 1) 语法：

```
if(boolean-1){
    语句块1
}else if(boolean-2){
    语句块2
}else if(boolean-3){
    语句块3
}else{
    语句块4
}
```

#### 2) 执行过程：

判断boolean-1，若为true则执行语句块1(整个结束)，若为false则再判断boolean-2，若为true则执行语句块2(整个结束)，若为false则再判断boolean-3，若为true则执行语句块3(整个结束)，若为false则执行语句块4(整个结束)

#### 3) 说明：

语句块1/2/3/4，必走其中之一-----多选一

```
package day04;

import java.util.Scanner;

//成绩等级判断
public class ScoreLevel {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("请输入成绩:");
    }
}
```

```

double score = scan.nextDouble();
//带数(888,-56,95,85,65,45)
if (score < 0 || score > 100) {
    System.out.println("成绩不合法");
} else if (score >= 90) {
    System.out.println("A-优秀");
} else if (score >= 80) { //默认已经小于90, 不用写
    System.out.println("B-良好");
} else if (score >= 60) {
    System.out.println("C-中等");
} else {
    System.out.println("D-不及格");
}
}
}

```

- switch...case: 多条路

优点: 效率高、结构清晰

缺点: 只能判断整数是否相等

break: 跳出switch

若业务是判断整数是否相等, 首选switch...case

```

package day04;

import java.util.Scanner;

//命令解析程序
public class CommandBySwitch {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("请选择功能: 1.取款 2.存款 3.查询余额 4.退卡");
        int command = scan.nextInt();

        switch (command) {
            case 1:
                System.out.println("取款业务...");
                break;
            case 2:
                System.out.println("存款业务...");
                break;
            case 3:
                System.out.println("查询余额业务...");
                break;
            case 4:
                System.out.println("退卡业务...");
                break;
            default:
                System.out.println("输入错误...");
        }
    }
}

```

3. 循环: 反复多次执行一段相同或相似的代码

4. 循环三要素:

- 循环变量的初始化
- 循环的条件(以循环变量为基础)
- 循环变量的改变(向着循环的结束改变)

循环变量：在整个循环过程中所反复改变的那个数

案例一：

输出5次"行动是成功的阶梯"

行动是成功的阶梯

行动是成功的阶梯

行动是成功的阶梯

行动是成功的阶梯

行动是成功的阶梯

循环变量：次数times

1) `int times=0;`

2) `times<5`

3) `times++;`

times=0/1/2/3/4/ 5时结束

案例二：

输出9的乘法表：

1\*9=9

2\*9=18

3\*9=27

4\*9=36

5\*9=45

6\*9=54

7\*9=63

8\*9=72

9\*9=81

循环变量：因数num

1) `int num=1;`

2) `num<=9`

3) `num++;`

num=1/2/3/4/5/6/7/8/9/ 10时结束

1\*9=9

3\*9=27

5\*9=45

7\*9=63

9\*9=81

循环变量：因数num

1) `int num=1;`

2) `num<=9`

3) `num+=2;`

num=1/3/5/7/9/ 11时结束

## 5. 循环结构：

- while结构：先判断后执行，有可能一次都不执行

1) 语法:

```
while(boolean){  
    语句块（反复执行的代码）  
}
```

2) 执行过程:

先判断boolean的值，若为true则执行语句块，  
再判断boolean的值，若为true则再执行语句块，  
再判断boolean的值，若为true则再执行语句块，  
如此反复，直到boolean的值为false时，while循环结束

```
package day04;
```

```
public class whileDemo {  
    public static void main(String[] args) {  
        //1)输出5次"行动是成功的阶梯":  
        int times = 0; //1.循环变量的初始化  
        while (times < 5) { //2.循环的条件  
            System.out.println("行动是成功的阶梯");  
            times++; //3.循环变量的改变  
        }  
        System.out.println("继续执行...");  
        /*  
            执行过程：（带数）  
                times=0  
            true  输出  times=1  
            true  输出  times=2  
            true  输出  times=3  
            true  输出  times=4  
            true  输出  times=5  
            false while循环结束  
            输出继续执行...  
        */  
  
        //2)输出9的乘法表:  
        int num = 1;  
        while (num <= 9) {  
            System.out.println(num + "*9=" + num * 9);  
            num += 2; //num++;  
        }  
        System.out.println("继续执行...");  
    }  
}
```

```
package day04;
```

```
import java.util.Scanner;
```

```
//猜数字游戏
```

```
public class Guessing {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
        int num = (int) (Math.random() * 1000 + 1); //1到1000之内的随机数  
        System.out.println(num); //作弊  
  
        //300(大),200(小),250(对)  
        System.out.println("猜吧!");  
    }  
}
```

```

    int guess = scan.nextInt(); //1.
    while (guess != num) { //2.
        if (guess > num) {
            System.out.println("太大了");
        } else {
            System.out.println("太小了");
        }
        System.out.println("猜吧!");
        guess = scan.nextInt(); //3.
    }
    System.out.println("恭喜你猜对了!");
}
}

```

- o do...while结构：先执行后判断，至少执行一次

当第1要素与第3要素相同时，首选do...while

1) 语法：

```

do{
    语句块
}while(boolean);

```

2) 执行过程：

先执行语句块，再判断boolean的值，若为true则  
再执行语句块，再判断boolean的值，若为true则  
再执行语句块，再判断boolean的值，若为true则  
再执行语句块，如此反复，直到boolean的值为false时，do...while循环结束

```

package day04;

import java.util.Scanner;

//猜数字游戏
public class Guessing {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int num = (int) (Math.random() * 1000 + 1); //1到1000之内的随机数
        System.out.println(num); //作弊

        //假设num=250
        //300(大),200(小),250(对)
        int guess;
        do {
            System.out.println("猜吧!");
            guess = scan.nextInt(); //1+3
            if (guess > num) {
                System.out.println("太大了");
            } else if (guess < num) {
                System.out.println("太小了");
            } else {
                System.out.println("恭喜你猜对了");
            }
        } while (guess != num); //2
    }
}

```

## 补充:

1. 任何复杂的业务逻辑都可以通过三种结构来实现:

- 顺序结构: 从上往下逐行执行, 每句必走
- 分支结构: 有条件的执行某语句一次, 并非每句必走
- 循环结构: 有条件的执行某语句多次, 并非每句必走

2. 随机生成:

```
int num = (int) (Math.random() * 1000 + 1); //1到1000之内的随机数
推导过程:
Math.random()-----0.0到0.999999999999999...
*1000-----0.0到999.9999999999999...
+1-----1.0到1000.9999999999999...
(int)-----1到1000
```

3. 变量的作用域/范围:

从变量的声明开始, 到包含它最近的大括号结束

4. 明日单词:

```
1)for:为了、循环的一种
2)continue:继续
3)result:结果
4)answer:回答
5)array/arr:数组
6)length:长度
7)multi:多
8)table:表格
9)addition:加法
10)index:下标、索引
11)out of:超出
12)bounds:界限
13)exception:异常
```

5. Scanner是java给大家提供的一个小零件, 包含了很多功能, 接收用户输入的数据只是它其中一个小功能而已。

6. 给变量赋值:

- 赋一个固定的值: int a = 250;
- 接收用户输入的值:

```
import java.util.Scanner;
Scanner scan = new Scanner(System.in);
int age = scan.nextInt();
```

- 系统随机生成值: Math.random() //0到0.999999...

## 晚课:

1. 补充易错点:

```
//编译器在编译的过程中, 用到变量时会去检查变量是否声明并初始化了
double score = 95.0;
String level;
```

```
// 声明level后, 要么初始化为"", 要么在所有分支中都给level赋值
if (score < 0 || score > 100) {
    level = "";
    System.out.println("成绩不合法");
} else if (score >= 90) {
    level = "A";
} else if (score >= 80) {
    level = "B";
} else if (score >= 60) {
    level = "C";
} else {
    level = "D";
}
```

## 2. 练习1:

```
int num = 9;
while (num >= 1) {
    System.out.println(num + "*9=" + num * 9);
    num--;
}
```

## 3. 练习2:

```
//1) 满1000打7折, 满500不满1000打8折, 不满500打9折
double price = 300.0;
if (price >= 1000) {
    price *= 0.7;
} else if (price >= 500) {
    price *= 0.8;
} else {
    price *= 0.9;
}
System.out.println("最终结算金额为:" + price);
```

## 扩展练习:

1. 接收用户输入的一个整数num, 判断它的正负零值, 正数则输出"+", 负数则输出 "-", 0则输出"0"

```
import java.util.Scanner;

public class extra_1 {
    public static void main(String[] args) {
        System.out.println("请输入一个整数: ");
        Scanner scan = new Scanner(System.in);
        int num = scan.nextInt();

        if (num > 0) {
            System.out.println("+");
        } else if (num < 0) {
            System.out.println("-");
        } else {
            System.out.println("0");
        }
    }
}
```

```
}
```

2. 接收用户输入的年份year和月份month，计算该年该月的天数，并输出

```
import java.util.Scanner;

public class extra_2 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("请输入年份: "); //代入1999, 2020, 2000
        int year = scan.nextInt();
        System.out.println("请输入月份: ");
        int month = scan.nextInt();
        int day = 0;

        switch (month) {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                day = 31;
                break;

            case 4:
            case 6:
            case 9:
            case 11:
                day = 30;
                break;

            case 2:
                //闰年: 能被4整除且不能被100整除, 或能被400整除
                if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
                    day = 29; //是闰年
                } else {
                    day = 28; //是平年
                }
                break;
        }

        System.out.println(year + "年" + month + "月有" + day + "天");
    }
}
```

3. 利用循环语句while计算数字1到100的和，并输出



```
public class extra_3 {  
    public static void main(String[] args) {  
        int num = 1, sum = 0;  
        while (num <= 100) {  
            sum += num;  
            num++;  
        }  
        System.out.println(sum);  
    }  
}
```