

语言基础第四天：

回顾：

1. 运算符：

- 算术：+、-、*、/、%、++、--
- 关系：>、<、>=、<=、==、!=
- 逻辑：&&、||、!
- 赋值：=、+=、-=、*=、/=、%=
- 字符串连接：+
- 条件：boolean?数1:数2

2. 分支结构(上)：基于条件执行的

- if结构：1条路
- if...else结构：2条路-----2选1
- if...else if结构：多条路-----多选1

精华笔记：

1. Scanner接收用户输入的数据：-----共3步，先背下来，面向对象第5天才能理解

2. 分支结构(下)：

- switch...case结构：多条路
 - 优点：效率高、结构清晰
 - 缺点：只能对整数判断相等
 - break：跳出switch

switch中数据的类型：byte,short,int,char,String,枚举

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：-----非常重要

- 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

循环变量：在整个循环过程中所反复改变的那个数

5. 循环结构：

- while结构：先判断后执行，有可能一次都不执行
- do...while结构：先执行后判断，至少执行一次

当第1要素的代码与第3要素的代码相同时，首选do...while

- for结构：应用率最高，适合与次数相关的

6. 三种循环结构如何选择：

- 先看循环是否与次数有关：

- 若有关-----直接上for
- 若无关，再看第1要素与第3要素的代码是否相同：
 - 若相同-----直接上do...while
 - 若不同-----直接上while

笔记：

1. Scanner接收用户输入的数据：-----共3步，先背下来，面向对象第5天才能理解

```
package day04;
import java.util.Scanner; //1. 导入一个扫描仪
//Scanner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，叫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+", 价格为:"+price);
    }
}
```

2. 分支结构(下):

- switch...case结构：多条路
 - 优点：效率高、结构清晰
 - 缺点：只能对整数判断相等
 - break：跳出switch

switch中数据的类型：byte,short,int,char,String,枚举

```
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.存款 2.取款 3.查询余额 4.退卡");
        int command = scan.nextInt();

        switch(command){
            case 1:
                System.out.println("存款业务...");
                break;
            case 2:
                System.out.println("取款业务...");
                break;
            //其他case...
        }
    }
}
```

```

        break;
    case 3:
        System.out.println("查询余额业务...");
        break;
    case 4:
        System.out.println("退卡成功");
        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：-----非常重要

- 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

循环变量：在整个循环过程中所反复改变的那个数

1) 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

2) 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/循环体-----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

```

//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:    3*9=27
int num = 1;
while(num<=9){
    System.out.println(num+"*9="+num*9);
    num++; //num+=2;
}
System.out.println("继续执行...");

```

```

package day04;
import java.util.Scanner;
//猜数字小游戏
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
    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("猜大了");
        }else{
            System.out.println("猜小了");
        }
        System.out.println("猜吧!");
        guess = scan.nextInt(); //3.
    }
    System.out.println("恭喜你猜对了!");
}
}

```

- do...while结构：先执行后判断，至少执行一次

当第1要素的代码与第3要素的代码相同时，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结束

```

package day04;
import java.util.Scanner;
//猜数字小游戏
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
        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("猜大了");
            }else if(guess<num){
                System.out.println("猜小了");
            }else{
                System.out.println("恭喜你猜对了");
            }
        }while(guess!=num);
    }
}

```

```

    }
    }while(guess!=num); //2
}
}

```

- o for结构：应用率最高，适合与次数相关的

1) 语法：

```

//    1    2    3
for(要素1;要素2;要素3){
    语句块/循环体-----反复执行的代码    4
}

```

2) 执行过程：

1243243243243243...2为false时，循环结束

//累加和:1+2+3+4+...+99+100=?

int sum = 0; //存和

for(int i=1;i<=100;i++){

sum = sum+i; //累加

}

System.out.println("sum="+sum);

/*

执行过程：

sum=0

i=1 sum=1

i=2 sum=1+2

i=3 sum=1+2+3

i=4 sum=1+2+3+4

... sum=1+2+3+4+...

i=99 sum=1+2+3+4+...+99

i=100 sum=1+2+3+4+...+99+100

i=101 false for循环结束

*/

//for中的循环变量num的作用域，仅在当前for中-----特殊记忆

for(int num=1;num<=9;num++){

System.out.println(num+"*9="+num*9);

}

//System.out.println(num); //编译错误，超出num作用域了

for(int num=1;num<=9;num+=2){

System.out.println(num+"*9="+num*9);

}

for(int times=0;times<5;times++){

System.out.println("行动是成功的阶梯");

}

System.out.println("继续执行...");

/*

执行过程：

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 for循环结束
输出继续执行...

*/
```

6. 三种循环结构如何选择：

- 先看循环是否与次数有关：
 - 若有关-----直接上for
 - 若无关，再看第1要素与第3要素的代码是否相同：
 - 若相同-----直接上do...while
 - 若不同-----直接上while

补充：

1. 任何复杂的程序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. 生成随机数：1到1000之内的

```
Math.random()-----0.0到0.9999999999...
*1000-----0.0到999.999999999...
+1-----1.0到1000.999999999...
(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. 变量的重名问题：

- 作用域重叠时，变量不同重名

5. 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
```

