

语言基础第五天：

笔记：

1. 循环结构：

- for结构：应用率最高，与次数相关的循环

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：

1243243243243243243...2（结束）

```
for (int times = 0; times < 5; times++) {
    System.out.println("行动是成功的阶梯");
}

for (int num = 1; num <= 9; num++) {
    System.out.println(num + "*9=" + num * 9);
}
//System.out.println(num); //报错
//for中的循环变量num的作用域，仅作用于当前for中

for (int num = 1; num <= 9; num += 2) {
    System.out.println(num + "*9=" + num * 9);
}
```

```
package day05;

import java.util.Scanner;

//随机加法运算器
public class Addition {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int score = 0;
        for (int i = 1; i <= 10; i++) { //10道题
            int a = (int) (Math.random() * 100); //0到99
            int b = (int) (Math.random() * 100);
            int result = a + b;
            System.out.println("(" + i + ") " + a + "+" + b + "=?");
            System.out.println("算吧！（输入-1退出程序）");
            int answer = scan.nextInt();
            if (answer == -1) { //选答案不可能出现的数，比如-1
                break;
            }
            if (answer == result) {
                System.out.println("答对了！");
                score += 10; //答对1题，加10分
            } else {

```

```

        System.out.println("答错了!"); //答错1题, 不扣分
    }
}
System.out.println("总分为:" + score);
}
}

```

补充:

```

//要素1放在循环外面也可以
int num = 1;
for (; num <= 9; num++) {
    System.out.println(num + "*9=" + num * 9);
}

//要素3放在循环里面也可以
for (int num1 = 1; num1 <= 9; ) {
    System.out.println(num1 + "*9=" + num1 * 9);
    num1++;
}

//要素1、2、3都没有, 即什么条件都没有, 是一个死循环
for (; ; ) {
    System.out.println("我爱Java");
}

//要素1、2、3也可以多个
for (int i = 1, j = 5; i <= 5 && j >= 1; i += 2, j -= 2) {
    System.out.println("i:" + i + " " + "j:" + j);
}

```

2. 三种结构如何选择:

- 先看循环是否与次数相关:
 - 若相关, for
 - 若无关, 再看要素1与要素3的代码是否相同:
 - 若相同, do...while
 - 若不同, while

3. break: 跳出循环

break用在switch中, 跳出分支。

break用在循环中, 跳出循环 (for、while、do..while) 。

```

for (int num = 1; num <= 9; num++) {
    if (num == 4) { //在某种特定情况下, 提前结束循环
        break; //跳出循环
    }
    System.out.println(num + "*9=" + num * 9);
}

```

continue: 跳过循环体中剩余语句, 进入下一次循环

continue只能用在循环中

```

//输出9的乘法表, 跳过能被3整除的

```

```

for (int num = 1; num <= 9; num++) {
    if (num % 3 == 0) {
        continue; //跳过循环体中剩余语句，进入下一次循环
    }
    System.out.println(num + "*9=" + num * 9);
}

//输出9的乘法表，只输出不能被3整除的
for (int num = 1; num <= 9; num++) {
    if (num % 3 != 0) {
        System.out.println(num + "*9=" + num * 9);
    }
}

//以上两段代码等价

```

4. 嵌套循环：

- 循环中套循环，常常多行多列时使用，一般外层控制行，内层控制列
- 执行规则：外层循环走一次，内层循环走所有次
- 建议：嵌套层数越少越好，能用一层就不用两层，能用两层就不用三层
- break只能跳出当前层循环

```

for (int num = 1; num <= 9; num++) { //控制行
    for (int i = 1; i <= num; i++) { //控制列
        System.out.print(i + "*" + num + "=" + i * num + "\t");
        //print(): 输出完不换行
        //" \t"控制输出的字符对齐
    }
    System.out.println(); //println()换行
}

```

5. 数组：

- 是一种数据类型（引用数据类型）
- 相同数据类型的元素的集合
- 声明：

```

//声明整型数组arr，包含10个元素，每个元素都是int型，默认值为0
int[] arr = new int[10];

```

- 初始化：（初始化的是数组中的元素）

```

int[] arr = new int[3]; //0,0,0
int[] arr1 = {1, 2, 3}; //1,2,3
int[] arr2 = new int[]{4, 5, 6}; //4,5,6

int[] arr3;
//arr3 = {7, 8, 9}; //编译错误，此方式只能在声明的同时初始化
arr3 = new int[]{7, 8, 9}; //正确

```

- 访问：（访问的是数组中的元素）
 - 通过 数组名.length 可以获取数组的长度（数组元素的个数）

```
int[] arr = new int[3];
System.out.println(arr.length); //3
```

- 通过下标/索引来访问数组中的元素，下标从0开始，最大到(数组的长度-1)

```
int[] arr = new int[3];
System.out.println(arr[0]); //输出第1个元素的值，0
arr[0] = 100; //给arr中的第1个元素赋值为100
arr[1] = 200; //给arr中的第2个元素赋值为200
arr[2] = 300; //给arr中的第3个元素赋值为300
//arr[3] = 400; //没有语法错误，不会标红，但运行时会发生数组下标越界异常
System.out.println(arr[arr.length - 1]); //输出最后一个元素的值
```

- 遍历/迭代：从头到尾挨个走一遍

```
int[] arr = new int[10];
for (int i = 0; i < arr.length; i++) { //遍历arr数组
    arr[i] = (int) (Math.random() * 100); //给每一个元素赋值为0到99的随机数
    System.out.println(arr[i]); //输出每个元素的值
}
```

补充：

1. 变量的重名问题：

- 当作用域重叠时，变量不能同名

2. \t：水平制表位，固定占8位，一般用于输出的内容对齐
3. 基本数据类型的默认值：

```
byte, short, int, long, char: 0
float, double: 0.0
boolean: false
```

4. 异常：

- `ArrayIndexOutOfBoundsException`：数组下标越界异常（数组下标为0到(数组长度-1)，超出范围则异常）

5. 明日单词：

```
1)copy:复制
2)arraycopy/copyOf:数组复制
3)max:最大值
4)min:最小值
5)sort:顺序、排序
6)method:方法
7)public static:公开静态的
8)void:空，没有返回结果的
9)return:返回
10)say:说
11)sayHi/sayHello:问好
12)getNum:获取数字
13)plus:加法
14)test:测试
```

扩展练习:

1. 利用for循环计算：求数字1到100之内，所有偶数的和，并输出

```
public class extra_1 {  
    public static void main(String[] args) {  
        int sum = 0;  
        for (int i = 2; i <= 100; i += 2) {  
            sum += i;  
        }  
        System.out.println(sum);  
    }  
}
```

2. 利用for循环计算：求8的阶乘，并输出

```
public class extra_2 {  
    public static void main(String[] args) {  
        int result = 1;  
        for (int i = 2; i <= 8; i++) {  
            result *= i;  
        }  
        System.out.println(result);  
    }  
}
```

3. 利用for循环计算：打印字符*组成的等腰三角形，输出结果如下所示：

```
  *  
 ***  
*****  
*****  
*****  
*****
```

```
public class extra_3 {  
    public static void main(String[] args) {  
        int row = 6; //行数  
        for (int i = 1; i <= row; i++) {  
            for (int j = 1; j <= row - i; j++) {  
                System.out.print(" ");  
            }  
            for (int j = 1; j <= i * 2 - 1; j++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```

