

语言基础第三天：

回顾：

1. 变量：存数的

```
int a;   int b,c,d,e;
int a = 250;   int a;   a=250;
int b = a+10;   System.out.println(b);
a = a+10; //在a本身基础之上增10
```

2. 八种基本数据类型：byte, short, int, long, float, double, boolean, char

- int: 整型, 4个字节, 5,25,250.....
- long: 长整型, 8个字节, 5L,1000000000000L.....
- double: 浮点型, 8个字节, 3.14,25.678.....
- boolean: 布尔型, 1个字节, true,false
- char: 字符型, 2个字节, '你','y','5','*'......

3. 类型间的转换：

- 两种方式：
 - 自动：小到大
 - 强制：大到小 (要转换成为的数据类型)变量 有可能溢出或丢失精度
- 两点规则：

```
short s1 = 5;
short s2 = 6;
short s3 = (short)(s1+s2);
```

精华笔记：

1. 运算符：

- 算术：+、-、*、/、%、++、--
- 关系：>、<、>=、<=、==、!=
- 逻辑：&&、||、!
- 赋值：=、+=、-=、*=、/=、%=
- 字符串连接：+
- 条件/三目：boolean?数1:数2

2. 分支结构：基于条件执行的语句

- if结构：1条路
- if...else结构：2条路

笔记：

1. 运算符：

○ 算术: +、-、*、/、%、++、--

- %:取模/取余, 余数为0即为整除

```
System.out.println(5%2); //1, 商2余1
System.out.println(8%2); //0, 商4余0----整除
System.out.println(2%8); //2, 商0余2
```

- ++/--:自增1/自减1, 可在变量前也可在变量后

- 单独使用时, 在前在后都一样
- 被使用时, 在前在后不一样
 - a++的值为a----- (a--的值为a)
 - ++a的值为a+1----- (--a的值为a-1)

```
//演示++单独使用
int a=5,b=5;
a++; //相当于a=a+1
++b; //相当于b=b+1
System.out.println(a); //6
System.out.println(b); //6

//演示++被使用
int a=5,b=5;
int c = a++; //1)将a++的值5赋值给c 2)a自增1变为6
//----a++的值为a(5), 所以c的值就是5
int d = ++b; //1)将++b的值6赋值给d 2)b自增1变为6
//++b的值为b+1(6), 所以d的值就是6
System.out.println(a); //6
System.out.println(b); //6
System.out.println(c); //5
System.out.println(d); //6

//演示--单独使用:
int a=5,b=5;
a--;
--b;
System.out.println(a); //4
System.out.println(b); //4

//演示--被使用:
int a=5,b=5;
int c = a--; //1)将a--的值5赋值给c 2)a自减1变为4
int d = --b; //1)将--b的值4赋值给d 2)b自减1变为4
System.out.println(a); //4
System.out.println(b); //4
System.out.println(c); //5
System.out.println(d); //4
```

○ 关系: >、<、>=、<=、==、!=

- 1)>(大于)、<(小于)
 - >=(大于或等于)、<=(小于或等于)
 - ==(等于)、!=(不等于)
- 2)关系运算的结果为boolean型，
关系成立则为true，关系不成立则为false

```
int a=5,b=10,c=5;
boolean b1 = a>b;
System.out.println(b1); //false
System.out.println(c<b); //true
System.out.println(a>=c); //true
System.out.println(a<=b); //true
System.out.println(a==c); //true
System.out.println(a!=c); //false
System.out.println(a+c>10); //false
System.out.println(a%2==0); //false
System.out.println(a++>5); //false-----a自增1变为6
System.out.println(a++>5); //true-----a自增1变为7
```

○ 逻辑：&&、||、!

- 逻辑运算是建立在关系运算的基础之上的
逻辑运算的结果也是boolean型
- &&:短路与(并且)，两边都为真则为真，见false则false

```
int a=5,b=10,c=5;
boolean b1 = b>=a && b<c;
System.out.println(b1); //true&&false=false
System.out.println(b<=c && b>a); //false&&true=false
System.out.println(a==b && c>b); //false&&false=false
System.out.println(b!=c && a<b); //true&&true=true

//演示短路
boolean b1 = a>b && c++>2;
System.out.println(b1); //false
System.out.println(c); //5，发生短路了
```

- ||:短路或(或者)，有真则为真，见true则true

```
int a=5,b=10,c=5;
System.out.println(b>=a || b<c); //true||false=true
System.out.println(b<=c || b>a); //false||true=true
System.out.println(b!=c || a<b); //true||true=true
System.out.println(a==b || c>b); //false||false=false

//演示短路
boolean b2 = a<b || c++>2;
System.out.println(b2); //true
System.out.println(c); //5，发生短路了
```

- !:逻辑非(取反)，非真则假，非假则真

```
int a=5,b=10,c=5;
boolean b2 = !(a<b);
System.out.println(b2);    ///!true=false
System.out.println(!(a>b)); ///!false=true
```

○ 赋值：=、+=、-=、*=、/=、%=

- 简单赋值运算符：=
- 扩展赋值运算符：+=、-=、*=、/=、%=
 - 注意:扩展赋值运算符自带强转功能

```
int a = 5;
a += 10; //相当于a=(int)(a+10)
System.out.println(a); //15
a *= 2; //相当于a=(int)(a*2)
System.out.println(a); //30
a /= 6; //相当于a=(int)(a/6)
System.out.println(a); //5

//小面试题:
short s = 5;
//s = s+10; //编译错误, 需强转: s=(short)(s+10);
s += 10; //相当于: s=(short)(s+10);
```

○ 字符串连接：+

- 若两边为数字，则做加法运算
- 若两边出现了字符串，则做字符串连接
- 任意类型的数据只要和字符串连，结果都会变为字符串类型---同化作用

```
int age = 38;
System.out.println("age="); //age=
System.out.println(age);    //38
System.out.println("age="+age); //age=38
System.out.println("我的年龄是"+age); //我的年龄是38
System.out.println("我今年"+age+"岁了"); //我今年38岁了

String name = "WKJ";
System.out.println("name="+name); //name=WKJ
System.out.println("大家好, 我叫"+name); //大家好, 我叫WKJ
System.out.println("大家好, 我叫"+name+", 今年"+age+"岁了"); //大家好, 我
    叫WKJ, 今年38岁了

System.out.println(10+20+"30"); //3030-----String
System.out.println(""+10+20+30); //102030-----String
System.out.println(10+20+30+""); //60-----String
```

○ 条件/三目：

- 语法：
 - boolean?数1:数2
- 执行过程：
 - 整个表达式是有值的，它的值要么是问号后的数1，要么是冒号后的数2
 - 判断boolean的值：

- 若为true, 则整个表达式的结果为?号后的数1
- 若为false, 则整个表达式的结果为:号后的数2

```
int num = 0;
int flag = num>0?1:-1;
System.out.println(flag); //-1

int a=8,b=55;
int max = a>b?a:b;
System.out.println("max="+max);
```

2. 分支结构: 基于条件执行的语句

- if结构: 1条路

1) 语法:

```
if(boolean){
    语句块-----基于条件执行的语句
}
```

2) 执行过程:

判断boolean的值:
 若为true, 则执行语句块(整个结束)
 若为false, 则直接结束

//1) 偶数的判断:

```
int num = 6;
if(num%2==0){ //带数(6,5)
    System.out.println(num+"是偶数");
}
System.out.println("继续执行...");
```

//1) 满500打8折

```
double price = 300.0; //消费金额
if(price>=500){ //满500    带数(600.0,300.0)
    price = price*0.8; //打8折
}
System.out.println("最终结算金额为:"+price);
```

- if...else结构: 2条路

1) 语法:

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

2) 执行过程:

判断boolean的值:
 若为true, 则执行语句块1(整个结束)
 若为false, 则执行语句块2(整个结束)

3) 说明:

语句块1和语句块2, 必走其中之一-----2选1

//1) 偶数、奇数的判断:

```
int num = 5; //带数(6,5)
```

```

if(num%2==0){
    System.out.println(num+"是偶数");
}else{
    System.out.println(num+"是奇数");
}
System.out.println("继续执行...");

//2)满500打8折, 不满500打9折:
double price = 300.0; //带数(600.0,300.0)
if(price>=500){
    price = price*0.8;
}else{
    price = price*0.9;
}
System.out.println("最终结算金额为:"+price);

```

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, 并非每句必走
 - 循环结构: -----明天讲
2. 明日单词:

```

1)Scanner/scan: 扫描仪
2)import: 引入、导入
3)System: 系统
4)in: 进入
5)new: 新的
6)nextInt: 下一个整数
7)nextDouble: 下一个浮点数
8)switch: 开关
9)case: 案例
10)break: 中断、退出
11)command: 命令
12)by: 通过
13)times: 次数
14)while: 当...的时候, 循环的一种
15)do: 做、干
16)math: 数字
17)random: 随机
18)guess/guessing: 猜
19)game: 游戏
20)level: 等级
21)count: 数量
22)score: 分数

```

