

第二章 变量与运算符

1. 变量（重点）

1.1 变量的概述

1.1.1 变量是什么

1. 变量是存储数据的容器
2. 现实生活中的容器：水杯，饭盒，储物柜，停车场
3. 计算机中的容器：硬盘，内存
4. 变量是在内存中开辟的一块空间

1.1.2 变量有什么用

1. 变量的作用：存，取数据
2. 和现实生活中的容器一样，用来存储特定的数据
3. 特定：约定好的

1.1.3 变量怎么用

- 变量的三要素

代码块

```
1  语法：数据类型    变量名称    = 值；  
2  举例：int         num         = 123；
```

1. 数据类型

a. 基本类型（四类八种）

i. 整型：存储整数

1. 先学习int类型，存储整数
2. 比如:123,789,10都是整数

ii. 浮点型：存储小数

iii. 字符型

iv. 布尔型

b. 引用类型（除了基本类型都是引用类型）

2. 变量名称

a. 自定义

b. 遵循命名规范

i. 规则：数字不可以开头，不可以使用java关键字和保留字

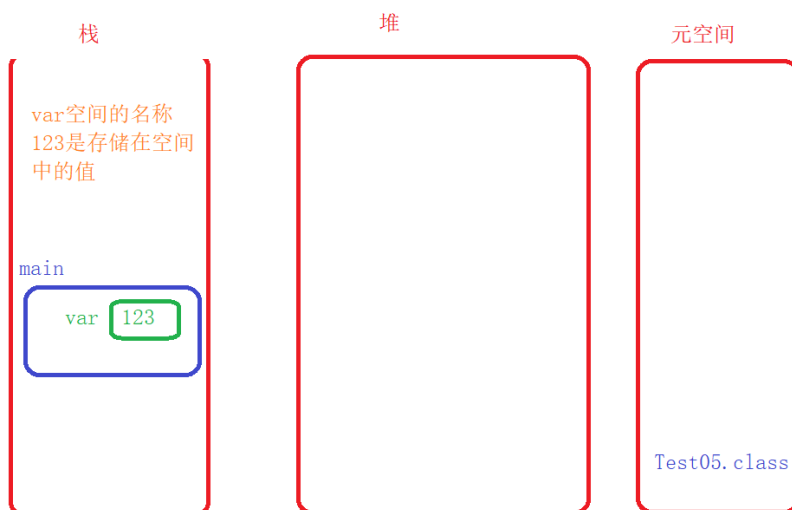
ii. 规范：变量和方法命名一样，首单词小写，其余单词首字母大写

3. 值：容器存储的数据

1.1.4 使用变量时的注意事项

1.2 容器的理解（掌握）

```
public class Test05 {  
    public static void main(String[] args) {  
  
        通过方法定义的变量称为：局部变量  
        局部变量，在栈中开辟内存  
  
        //数据类型 变量名称 = 值;  
        int var = 123;  
    }  
}
```



1.3 变量的语法结构（掌握）

1.3.1 语法一：变量的声明并赋值

1. 语法：数据类型 变量名称 = 值;

2. 举例：

代码块

```
1 public class Test06 {  
2     public static void main(String[] args) {  
3         /**  
4             * 1.变量的声明并赋值  
5             * 2.语法：数据类型 变量名称 = 值;
```

```

6          *    1.数据类型：约束空间存储的数据类型
7          *    2.变量名称：空间的名称
8          */
9          int var = 123;
10         System.out.println(var);
11         //变量定义后可以多次使用
12         System.out.println(var);
13     }
14 }

```

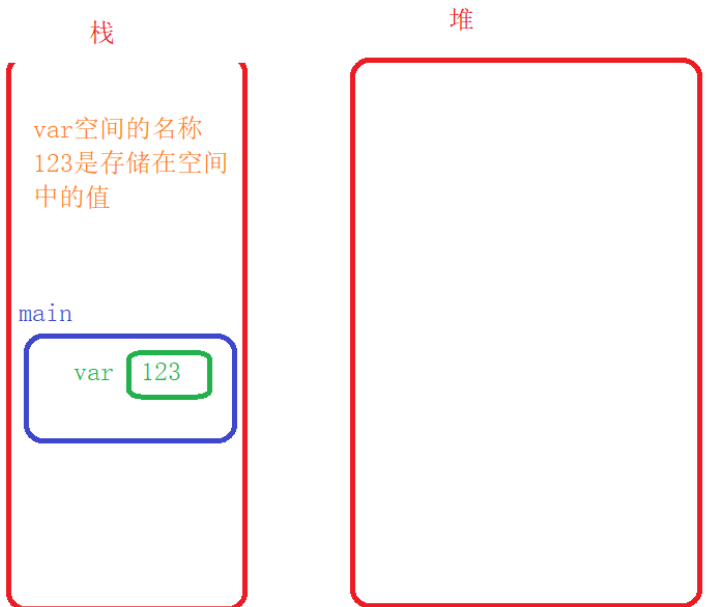
```

public class Test06 {
    public static void main(String[] args) {
        /**
         * 1. 变量的声明并赋值
         * 2. 语法：数据类型 变量名称 = 值;
         *    1. 数据类型：约束空间存储的数据类型
         *    2. 变量名称：空间的名称
         */
        int var = 123;
        System.out.println(var);
    }
}

```

输出var:

1. 找到var命名的那块空间
2. 取出空间的数据



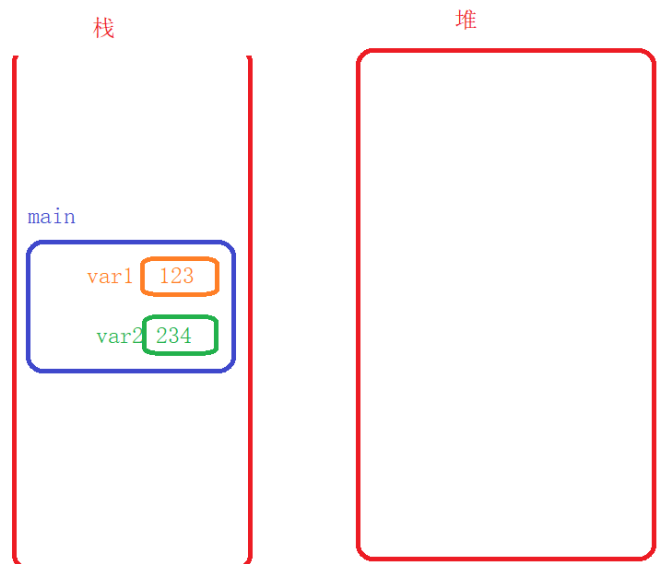
1.3.2 语法二：变量先声明，后赋值

```

public class Test07 {
    public static void main(String[] args) {
        /**
         * 语法:
         * 1. 声明：数据类型 变量名称
         * 2. 赋值：变量名称 = 值;
         * 注意：变量，必须先声明，后赋值，再使用
         */
        int var1 ;
        var1 = 123;
        //Variable 'var' might not have been initialized
        System.out.println(var1);

        int var2 = 234;
        System.out.println(var2);
    }
}

```



```

1  public class Test07 {
2      public static void main(String[] args) {
3          /**
4           * 语法:
5           * 1.声明: 数据类型 变量名称
6           * 2.赋值: 变量名称 = 值;
7           * 注意: 变量, 必须先声明, 后赋值, 再使用
8           */
9          int var1 ;
10         var1 = 123;
11         //Variable 'var' might not have been initialized
12         System.out.println(var1);
13
14         int var2 = 234;
15         System.out.println(var2);
16     }
17 }

```

1.3.3 语法三：声明并赋值多个变量

代码块

```

1  public class Test08 {
2      public static void main(String[] args) {
3          //语法: 数据类型 变量名称1 = 值1,变量名称2 = 值2;
4          int var1 = 2,var2 = 3;
5          System.out.println(var1);//2
6          System.out.println(var2);//3
7
8          var1 = 6;//修改var1空间的值
9
10         System.out.println(var1);//6
11         System.out.println(var2);//3
12
13     }
14 }

```

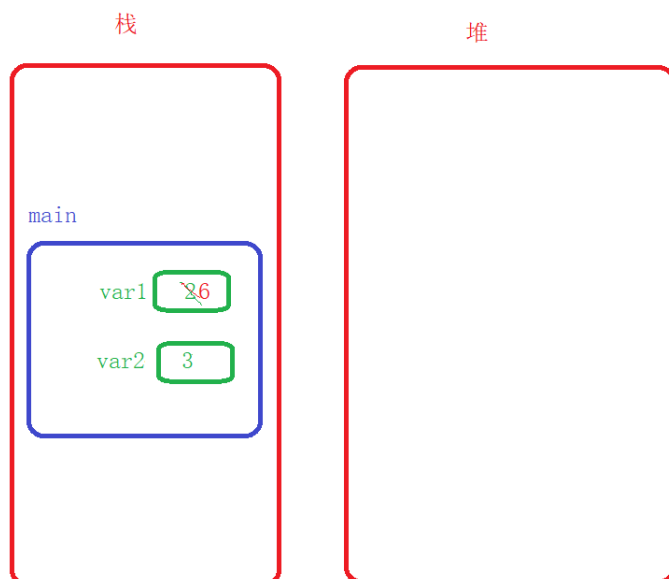
```

public class Test08 {
    public static void main(String[] args) {
        //语法: 数据类型 变量名称1 = 值1, 变量名称2 = 值2;
        int var1 = 2, var2 = 3;
        System.out.println(var1); //2
        System.out.println(var2); //3

        var1 = 6; //修改var1空间的值

        System.out.println(var1); //6
        System.out.println(var2); //3
    }
}

```



1.3.4 语法四：声明多个变量，挨个赋值

代码块

```

1  public class Test09 {
2      public static void main(String[] args) {
3          /**
4           * 1.声明多个变量
5           *    数据类型 变量名称1, 变量名称2
6           * 2.挨个赋值
7           *    变量名称1 = 值1;
8           *    变量名称2 = 值2;
9           */
10         int var1, var2 ;
11         var1 = 2;
12         var2 = 3;
13         System.out.println(var1);
14         System.out.println(var2);
15     }
16 }

```

1.4 代码格式的要求

1. 遵循遇到{} 缩进tab
2. 遇到运算符，左右添加空格

代码块

```

1  public class Test10 {

```

```

2      public static void main(String[] args) {
3          int var = 123;
4          System.out.println(var);
5      }
6  }

```

3. Ctrl + alt + L :代码格式化

2. 运算符（掌握）

2.1 二元运算符

- 加 (+) , 减 (-) , 乘 (*) , 除 (/) ,取余(%)

代码块

```

1  public class Test11 {
2      public static void main(String[] args) {
3          //加 (+) , 减 (-) , 乘 (*) , 除 (/) ,取余(%)
4          int x = 10;
5          int y = 3;
6          System.out.println(x + y); //13
7          System.out.println(x - y); //7
8          System.out.println(x * y); //30
9          System.out.println(x / y); //3, int 类型只能存储整数, 小数部分-舍弃
10         System.out.println(x % y); //1 取余数
11     }
12 }

```

2.2 一元运算符

- 正 (+) , 负 (-) , 前后递增 (++) , 前后递减 (--)

代码块

```

1  public class Test12 {
2      public static void main(String[] args) {
3          //正 (+) , 负 (-) , 前后递增 (++) , 前后递减 (--)
4          int x = 2 , y = 3;
5          System.out.println(++x); //2
6          System.out.println(-y); // -3
7          System.out.println("-----");
8
9          /**
10         * 前后递增 (++)

```

```

11      * 1. ++ 自增1, 如果变量为2, 那么自增1后的结果是3
12      * 2. ++ 又分为前 ++ 和 后 ++
13      *     1.前++, ++在变量的前面
14      *     2.后++, ++在变量的后面
15      * 3.前++和后++的区别
16      *     1.前++, 先自增1, 后运算或者输出
17      *     2.后++, 先运算或者输出, 后自增1
18      *
19      */
20      int m = 2, n = 3;
21      //m = m + 1;
22      System.out.println(++m); //3
23      System.out.println(m); //3
24      System.out.println(n++); //3
25      System.out.println(n); //4
26      System.out.println("-----");
27      int i = 2;
28      int j = 3;
29      int z = ++i; //i = i + 1
30      System.out.println(z); //3
31      System.out.println(i); //3
32
33      z = i ++;
34      System.out.println(z); //3
35      System.out.println(i); //4
36      /**
37       * 伪代码
38       *     int temp = i; //把i的值 4 赋值给 temp
39       *     i = i + 1; i = 5;
40       *     i = temp ; 把临时变量的值赋值给了i, 所以最终i的值还是 4
41       */
42      i = i ++;
43      System.out.println(i); //4
44
45      z = i++ + 1;
46      /**
47       * 执行过程:
48       *     1.i + 1 先运算, 4 + 1
49       *     2.把i + 1的结果赋值给z
50       *     3.i 自增1, i = 5
51       */
52      System.out.println(z); //5
53      System.out.println(i); //5
54
55      z = ++i + 1;
56      /**
57       * 执行过程

```

```

58      *   1.i先自增1,i = 6
59      *   2.i + 1 ,6 + 1 : 7
60      *   3.把7赋值给z
61      */
62      System.out.println(z);//7
63      System.out.println(i);//6
64
65      i = 5;
66      int num = ++i + i++ * --i;
67      /**
68      * 分步计算:
69      *   1.++i ; i 先变为 6, 此时表达式: 6 + i++ * --i;
70      *   2.i++ ; 先使用i的值参与乘法运算, 然后i再变为7, 此时表达式: 6 + 6++
71      *   3.--i ; i先变为6, 然后再参与乘法运算: 此时表达式 6 + 6 * 6;
72      *   4.最终 6 + 36 = 42
73      *   5.i最终 = 6
74      */
75      System.out.println(num);//42
76      System.out.println(i);//6
77
78
79      }
80      }

```

2.3 赋值运算符

- 直接赋值 (=) , 复合赋值 (+=, -=, *=, /=,%=)

代码块

```

1  public class Test13 {
2      public static void main(String[] args) {
3          // 直接赋值 (=) , 复合赋值 (+=, -=, *=, /=,%=)
4          int x = 2;
5          //x = x + 3;
6          x+=3;//等同于 x = x + 3;
7          System.out.println(x);//5
8
9          int y = 2;
10         y -= 1;
11         System.out.println(y);//1
12
13         int m = 3;
14         m *= 2;
15         System.out.println(m);//6
16

```



```
17         int n = 10;
18         n /= 3;
19         System.out.println(n); //3
20
21         int z = 10;
22         z %= 3;
23         System.out.println(z); //1
24
25
26     }
27 }
```

作业

1. 练习题（注释）：

- 声明Exercise01类，打印输出正方形如下：

代码块

```
1  *****
2  *****
3  *****
```

- 在其中适当的位置加入单行注释和多行注释，以增加代码的可读性。编译并运行程序，验证注释对代码执行是否有影响

2. 练习题（变量）

编写一个Exercise02类，main方法中声明两个变量var1和var2并赋值，声明var3变量，保存var1和var2之积，然后打印输出var3的值。

3. 练习题（复合运算）

声明Exercise03类，main方法中声明var1,var2并赋值，然后用var2保存var2与var1之和，打印输出var2的值。分别用（+和+=）两种方式计算。

4. 练习题

声明变量并赋值的四种语法，都练习一下

5. 练习题（前--，和后--，通过做练习自己总结）

声明两个变量var1 和 var2并赋值

运用var1 和var2 测试前-- 和后--的不同，并总结

6. 把上课的代码都敲一遍

7. 预习

3.