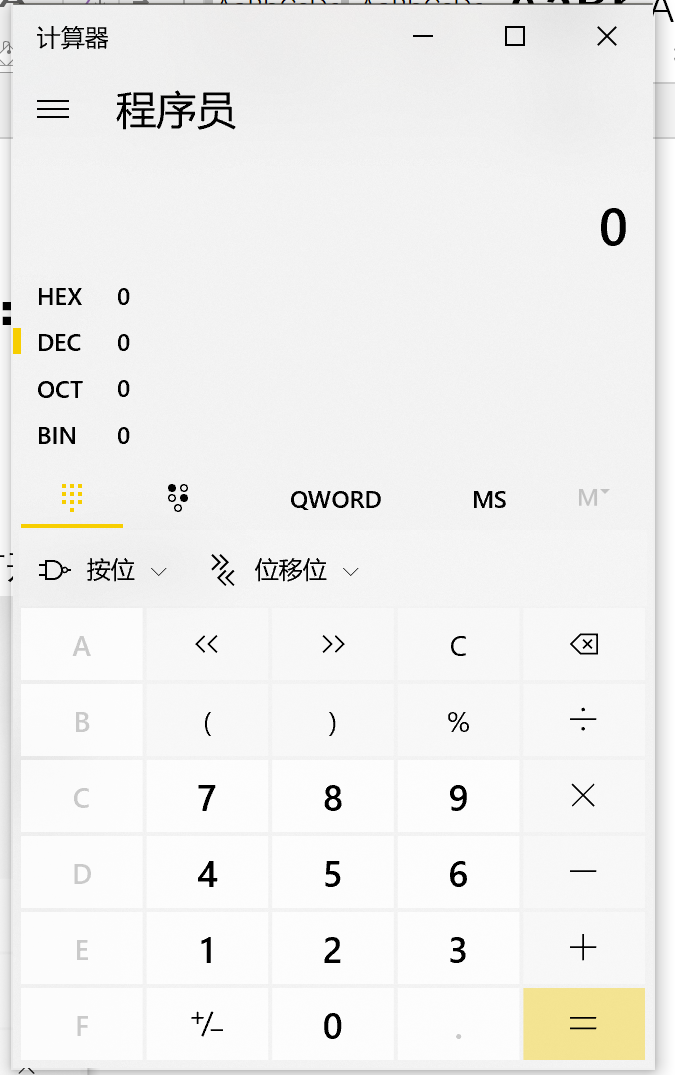
# 黑马Java

## Day 01

### 计算机基础知识：

### 进制转换

电脑自带的计算，可以打开程序员模式，HEX：16进制，DEC：十进制，OCT：八进制，BIN：2进制，可以在各个进制进行切换和计算。



### 常用DOS命令：

1. 如何在当前目录打开CMD

长按Shift，鼠标右键点击当前目录空白处，先打开Powershell窗口，输入 start cmd 回车

1. 

### HelloWorld小程序编译执行

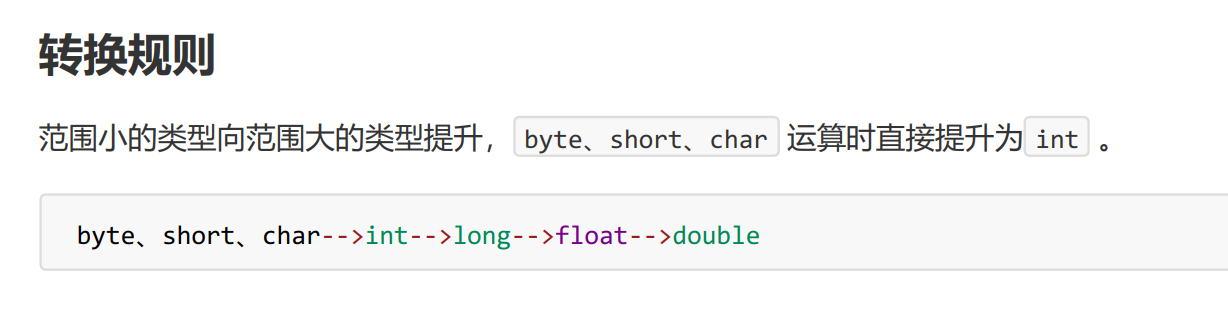
* 新建HelloWorld小程序，文件为HelloWorld.java
* 在当前目录打开cmd，运行javac HelloWorld.java，命令行无提示，但是在当前目录下生成了HelloWorld.class文件，该文件该文件就是编译后的文件，是Java的可运行文件，称为字节码文件，有了字节码文件，就可以运行程序了。
* cmd中使用命令 java HelloWorld 运行该程序

## Day 02

### 数据类型转换：

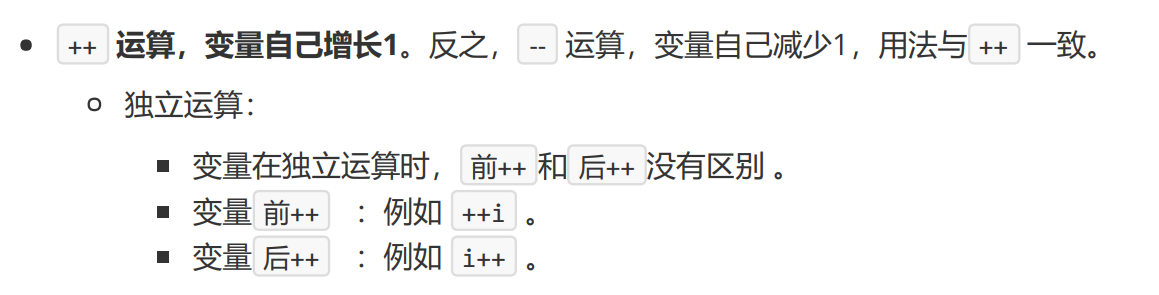
Java程序中要求参与的计算的数据，必须要保证数据类型的一致性，如果数据类型不一致将发生类型的转换

自动转换：将取值范围小的类型自动提升为取值范围大的类型 。

强制转换：将取值范围大的类型强制转换成取值范围小的类型 。（手工）



### ++运算和--运算





## Day 03

### 循环语句：

switch语句使用的注意事项：

1. 多个case后面的数值不可以重复。

2. switch后面小括号当中只能是下列数据类型：

基本数据类型：byte/short/char/int

引用数据类型：String字符串、enum枚举

3. switch语句格式可以很灵活：前后顺序可以颠倒，而且break语句还可以省略。

“匹配哪一个case就从哪一个位置向下执行，直到遇到了break或者整体结束为止。”

## Day 04

### IDEA快捷键：

自动生成代码：

* psvm 回车: public static void main(String[] args){}
* sout 回车：System.out.println();
* for循环代码生成：循环次数.for 回车，例如

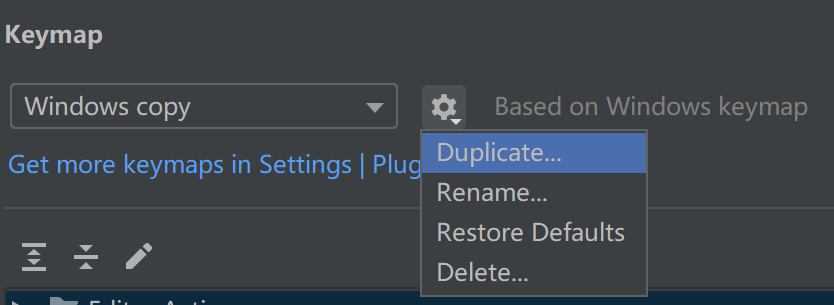
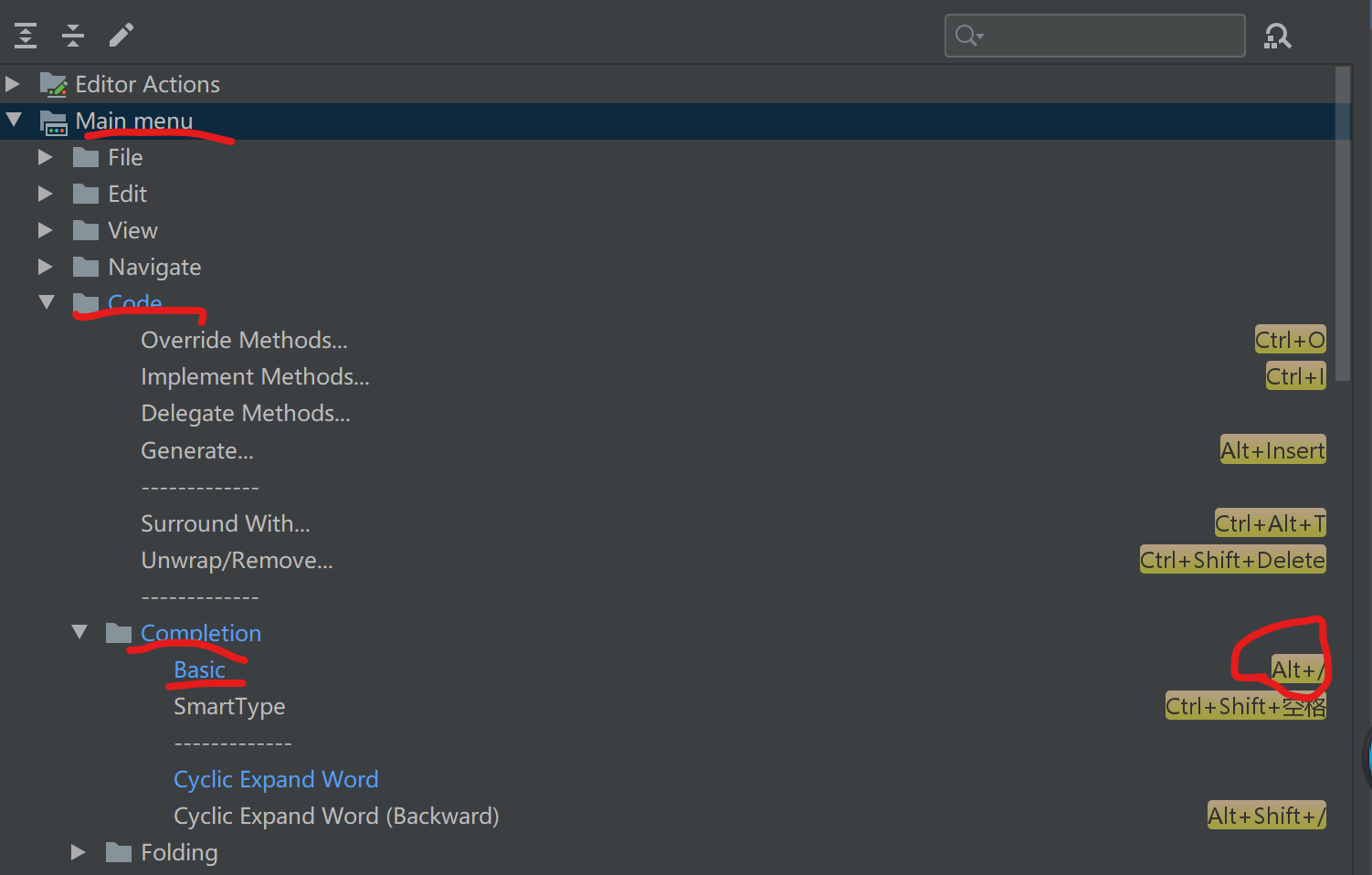
5.for 回车：for (int i = 0; i < 5; i++) { }

Alt 4：调出java编译执行结果界面

更改某些设置：

字体：settings🡪Eitor🡪font

快捷键补全设置：settings🡪Keymap🡪Duplicate🡪Main meun🡪Code🡪Comption🡪Basic

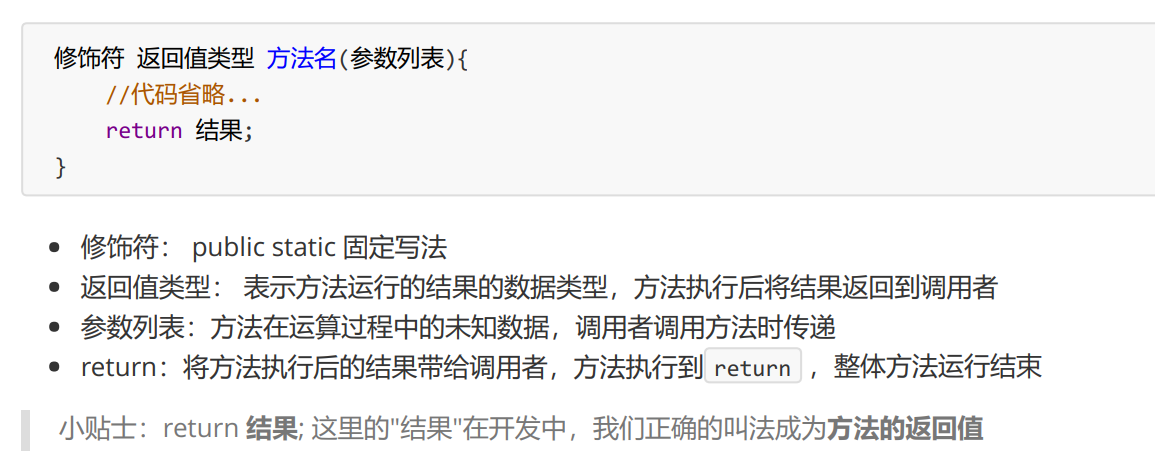


Alt+Enter:将光标放在报错位置，Alt+Enter，将会提出一些改错方式，尝试进行错误修复

Ctrl+Alt+L：自动对齐代码

同一项目下新建Module

### 定义方法的格式 ：



定义位置，类中方法外面。

返回值类型，必须要和 return 语句返回的类型相同，否则编译失败 。

### 调用方法格式



System.out.print("XXX"); 打印XXX

System.out.println("XXX"); 打印XXX并换行

### 方法重载：

方法重载：指在同一个类中，允许存在一个以上的同名方法，只要它们的参数列表不同即可，与修饰符和返回值类型无关。

参数列表：个数不同，数据类型不同，顺序不同。

重载方法调用：JVM通过方法的参数列表，调用不同的方法。

## Day 05

### 数组定义格式：

* 数组存储的数据类型[] 数组名字 = new 数组存储的数据类型[长度]；

int[] arr = new int[3];

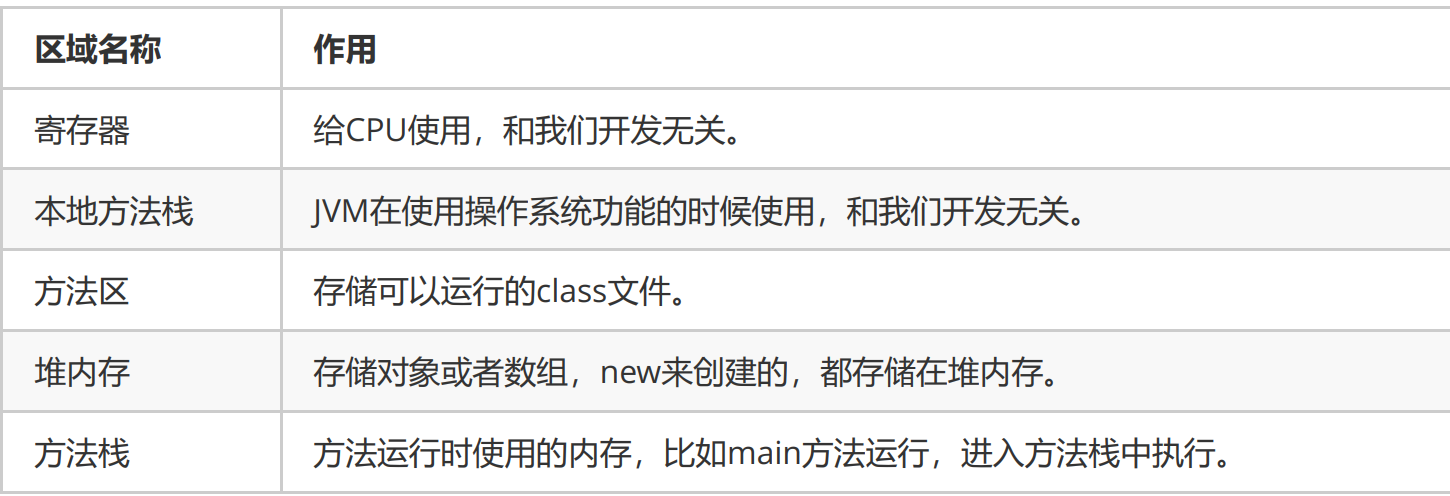
* 数据类型[] 数组名 = new 数据类型[]{元素1,元素2,元素3...};

int[] arr = new int[]{1,2,3,4,5};

* 数据类型[] 数组名 = {元素1,元素2,元素3...};

int[] arr = {1,2,3,4,5};

### JVM内存划分



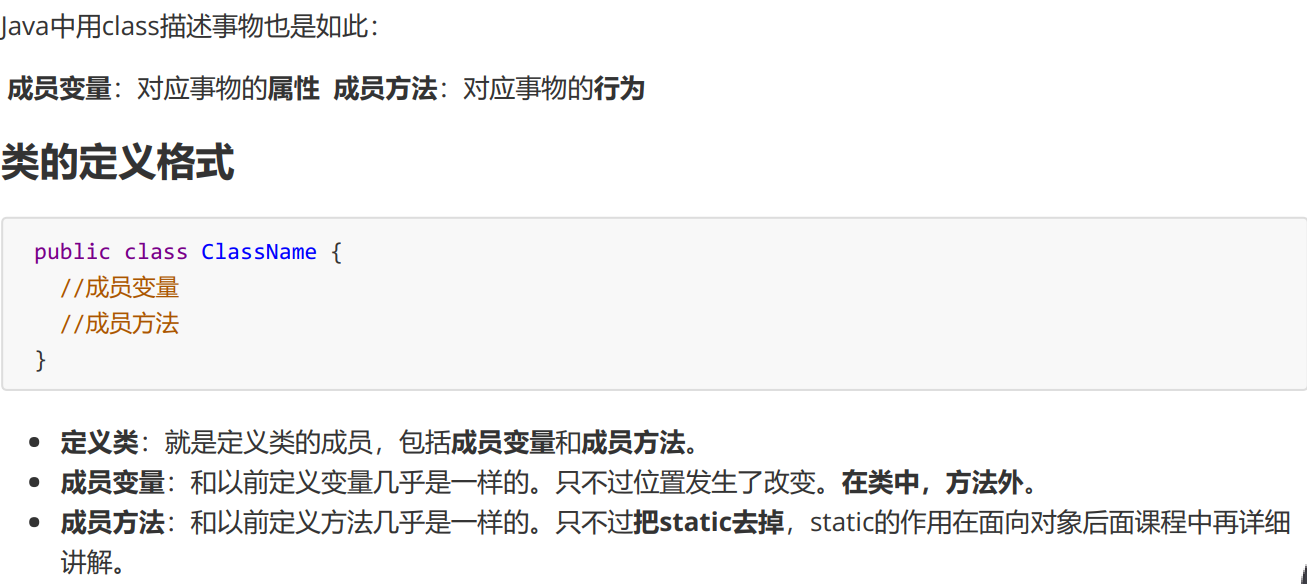
## Day 06

### 类与对象

* 类：是一组相关属性和行为的集合。可以看成是一类事物的模板，使用事物的属性特征和行为特征来描述该类事物。现实中，描述一类事物：
* 属性：就是该事物的状态信息。
* 行为：就是该事物能够做什么。
* 对象：是一类事物的具体体现。对象是类的一个实例（对象并不是找个女朋友），必然具备该类事物的属性和行为。

现实中，一类事物的一个实例

### 类的定义



### 类的使用

通常，一个类不能直接使用，需要根据类创建一个对象。使用步骤如下：

1. 导包：指出所需要用的类的位置（同一个包下可以省略导包步骤）

import 包名称.类名称

1. 创建对象格式：

类名 对象名 = new 类名();（new 之后，没有赋值的成员变量为初始值）

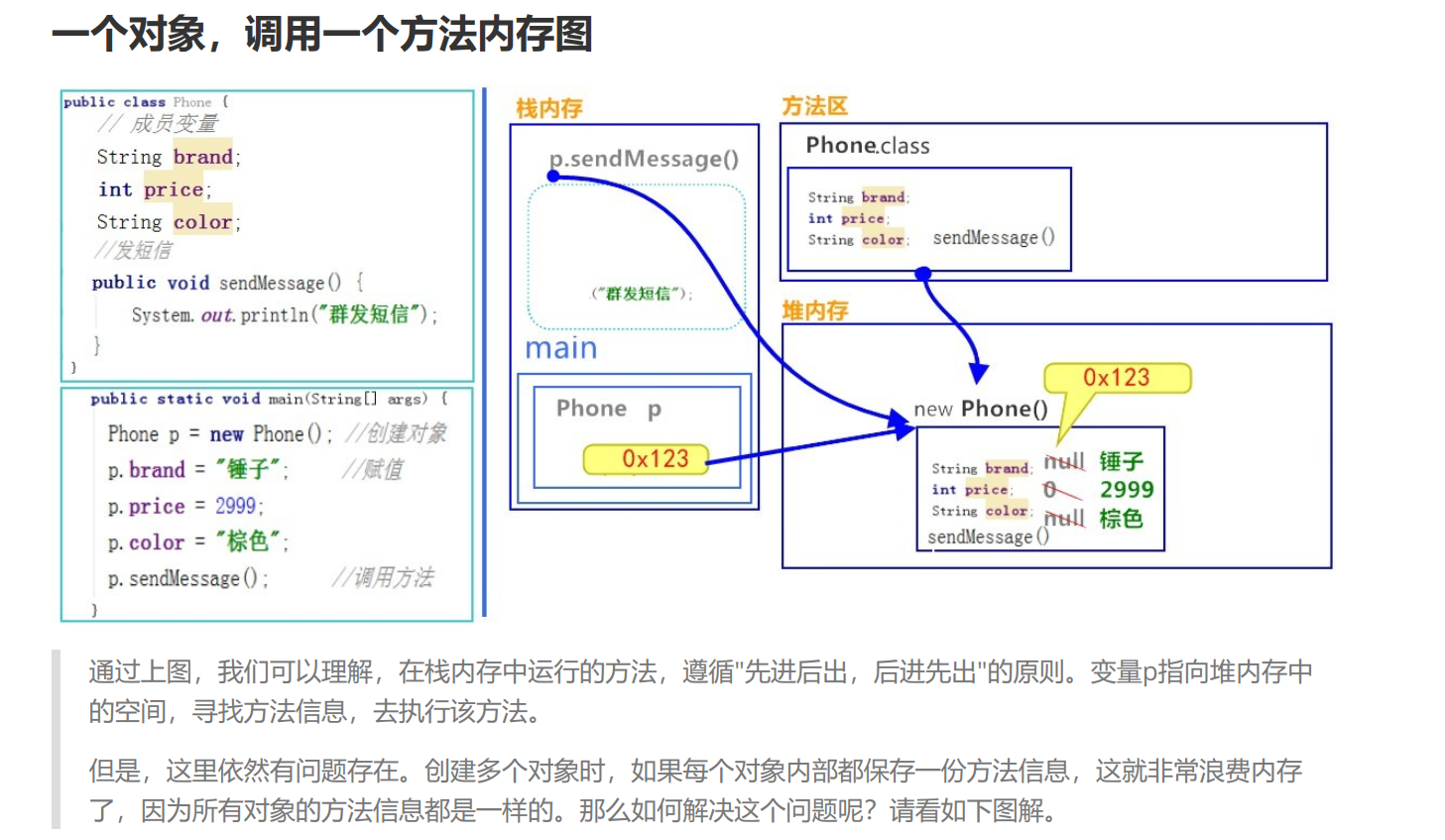


1. 使用类的成员变量或成员方法

对象名.成员变量；

对象名.成员方法([参数])；

### 对象内存图



方法区内存储着类成员变量和成员方法，其中，成员方法有自己的地址A；运行中的方法存储在栈中，栈先进后出（压栈），程序执行从Main开始，因此，Main先进入栈，执行到new对象时，new 对象进入堆，此时 ，栈中存储对象在堆中的地址B；堆中存储初始化后的成员变量，以及成员方法的地址A；栈中继续执行对成员变量进行操作，操作 成员方法时，由B🡪A,找到该方法，执行完成后出栈。

多个不同对象（new 多个对象 ）也是同样的方法。

引用类型作为参数，传递的是地址值。

### 成员变量和局部变量区别

* 在类中的位置不同 重点

成员变量：类中，方法外

局部变量：方法中或者方法声明上(形式参数)

* 作用范围不一样 重点

成员变量：类中

局部变量：方法中

* 初始化值的不同 重点

成员变量：有默认值

局部变量：没有默认值。必须先定义，赋值，最后使用

* 在内存中的位置不同 了解

成员变量：堆内存

局部变量：栈内存

* 生命周期不同 了解

成员变量：随着对象的创建而存在，随着对象的消失而消失

局部变量：随着方法的调用而存在，随着方法的调用完毕而消失

### 封装

原则

将属性隐藏起来，若需要访问某个属性，提供公共方法对其访问。

封装的步骤

1. 使用 private 关键字来修饰成员变量。

2. 对需要访问的成员变量，提供对应的一对 getXxx 方法 、 setXxx 方法

注意：getXxx方法不能有传入参数，返回值类型和成员变量对应

SetXxx方法不能有返回值，传入参数和成员变量对应

若成员变量是Boolean类型，setXxx不变，对应的是isXxx,用法 同getXxx

### this关键字：在重名时，进行区分

当局部变量和成员变量重名时，默认就近原则，优先使用局部变量

如果需要访问本类中成员变量，使用关键词this.成员变量名

通过谁调用的方法，谁就是this，例如，person.getName()方法中，person调用的方法，则person就是this。

### 构造方法

用new关键字新建一个对象时，实际上就是使用该类的构造方法是 。

注意事项

1. 如果你不提供构造方法，系统会给出无参数构造方法。

2. 如果你提供了构造方法，系统将不再提供无参数构造方法。

3. 构造方法是可以重载的，既可以定义参数，也可以不定义参数。



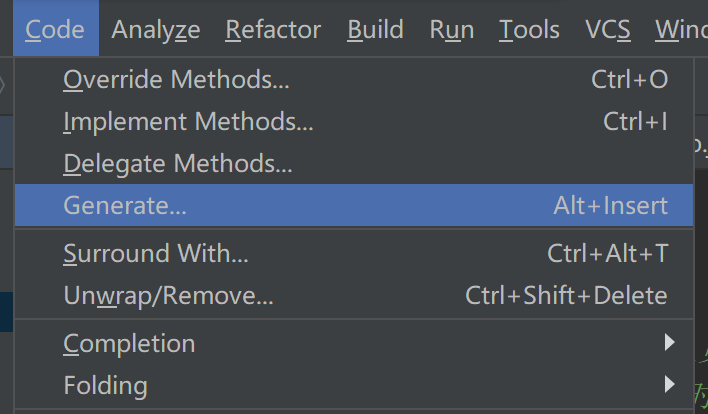
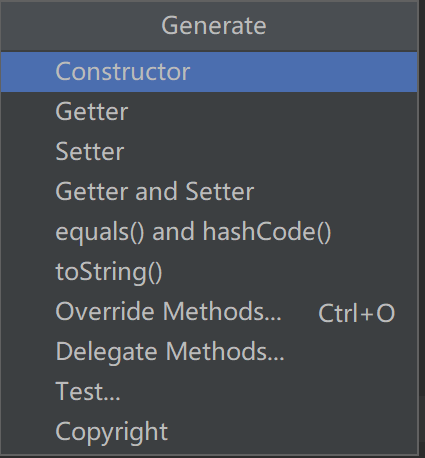
### 定义一个标准的类：JavaBean

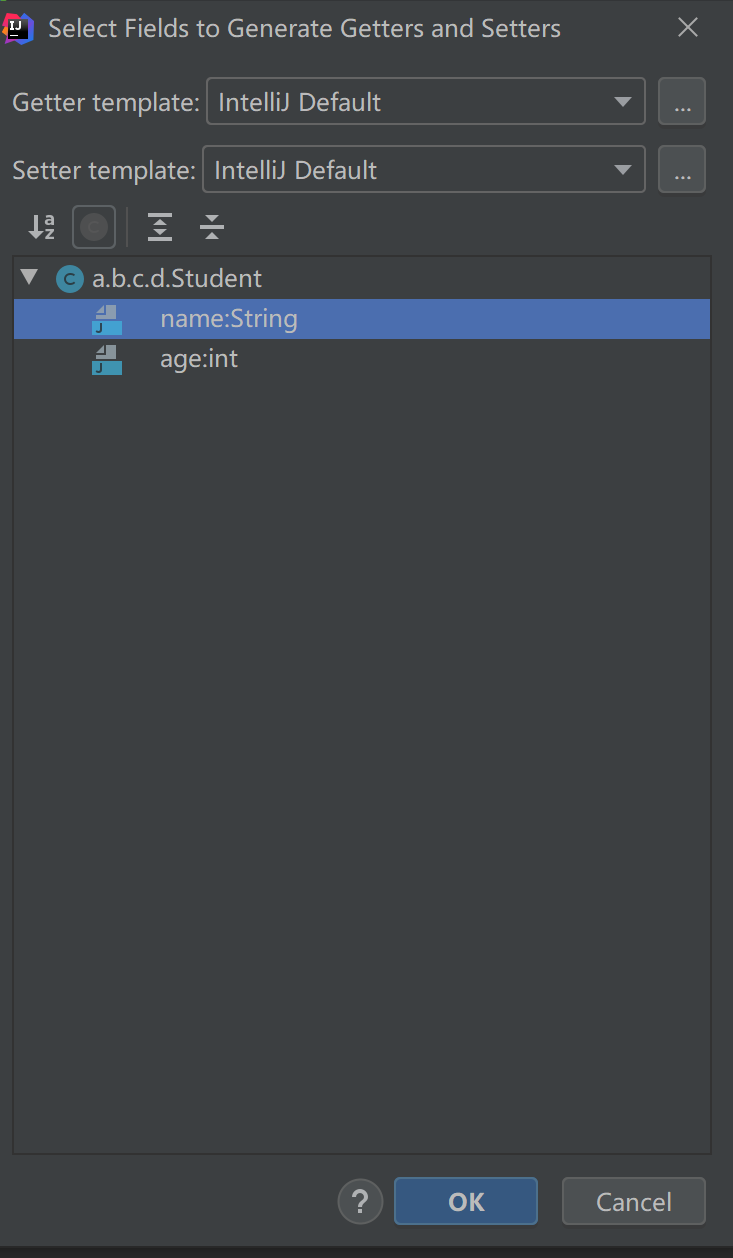
要求:

* 所有成员变量用private修饰
* 为每一个成员变量 编写 getter/setter方法
* 编写一个无参数构造方法
* 编写一个全参数构造方法

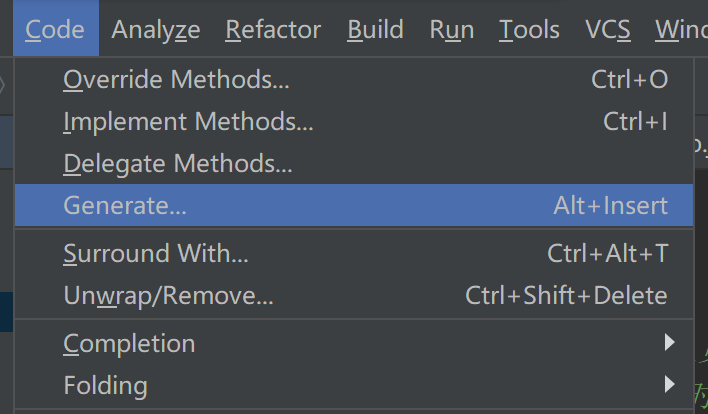
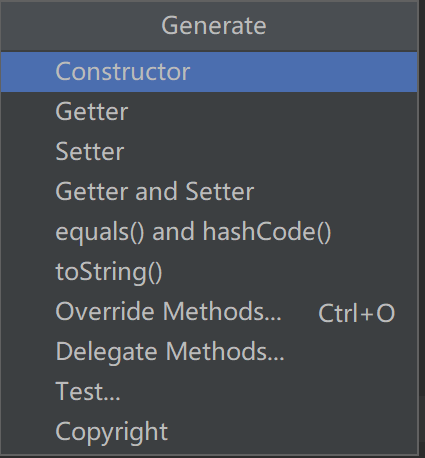
### getter/setter代码自动生成 ：

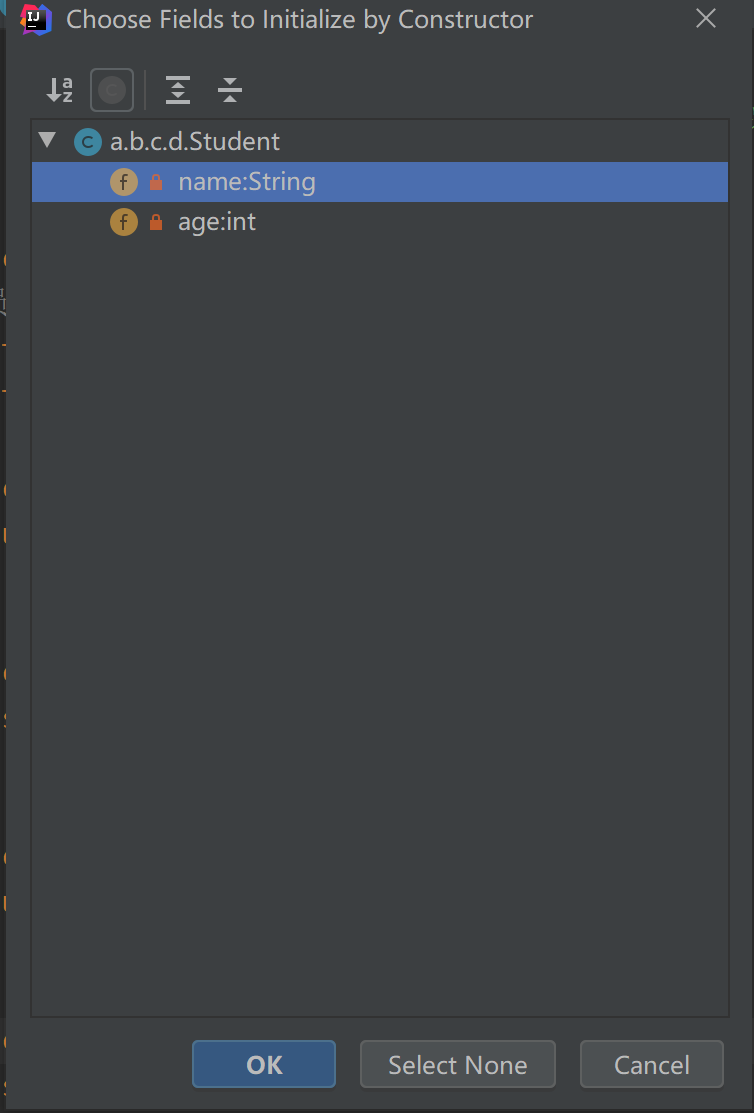
编写完private成员变量后 ，可以自动生成getter/setter方法，步骤如下：

  点击getter and setter

按住Shift选中想要的成员变量，点击OK即可

### 无参/全参数构造方法自动生成：

  选择Constructor

 无参构造方法直接点击Select None即可生成；全参构造方法按Shift多选，选择全部参数 ，点击 OK即可

## Day 07

API使用步骤

1. 打开帮助文档。

2. 点击显示，找到索引，看到输入框。

3. 你要找谁？在输入框里输入，然后回车。

4. 看包。java.lang下的类不需要导包，其他需要。

5. 看类的解释和说明。

6. 学习构造方法。

7. 使用成员方法

API文档

### 引用类型一般使用步骤：

* 导包

使用import关键字导包，在类的所有代码之前导包，引入要使用的类型，java.lang包下的所有类无需导入。 格式：import 包名.类名;

* 创建对象

使用该类的构造方法，创建一个该类的对象。 格式：数据类型 变量名 = new 数据类型(参数列表);

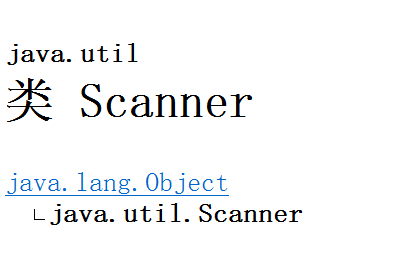
* 调用方法

调用该类的成员方法，完成指定功能。 格式：变量名.方法名();

### Scanner类

使用方法：

1. 导包：



import java.util.Scanner;

1. 创造类：

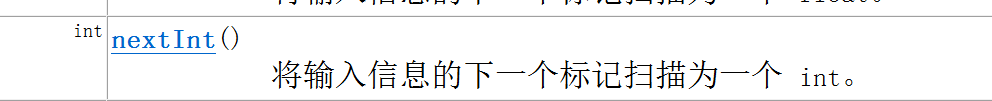


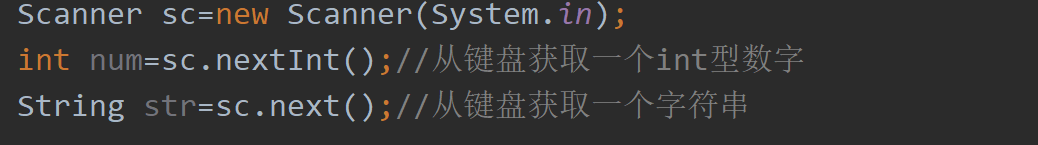
例如：

Scanner sc=new Scanner(System.in);

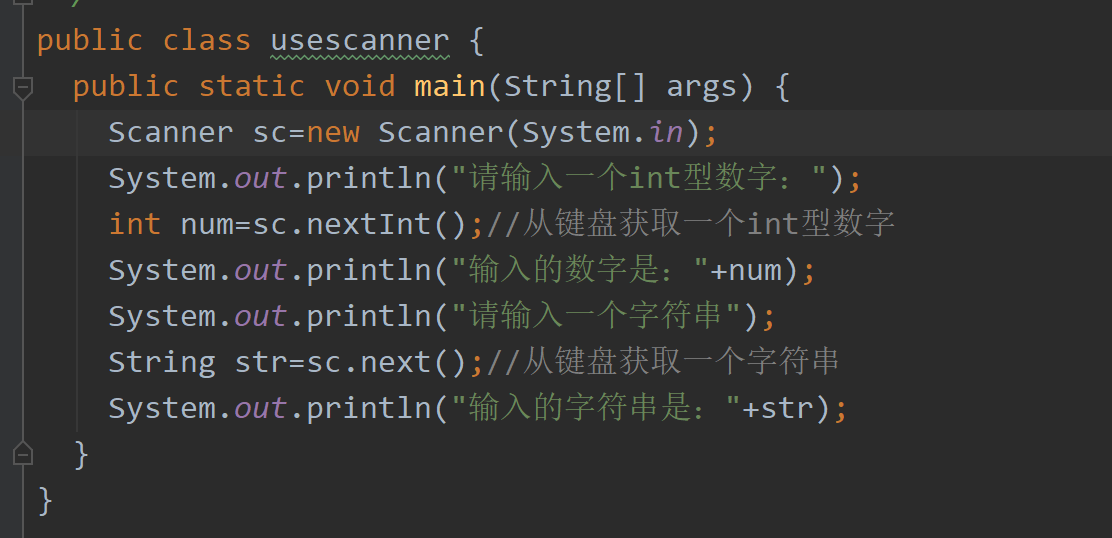
System.in表示从键盘输入

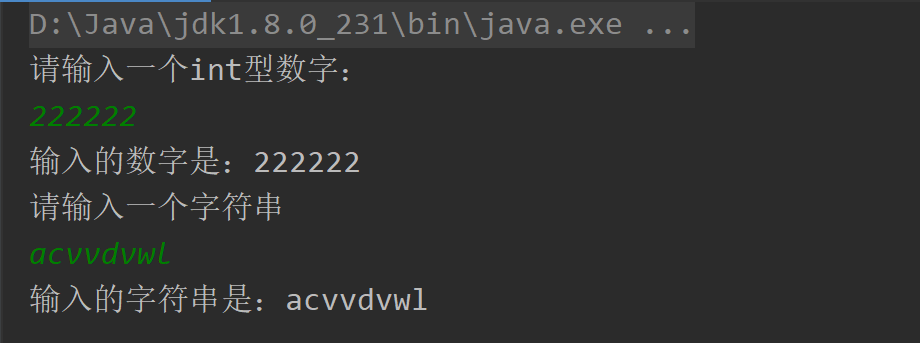
1. 使用，调用API中的成员方法，查阅API文档，例如：





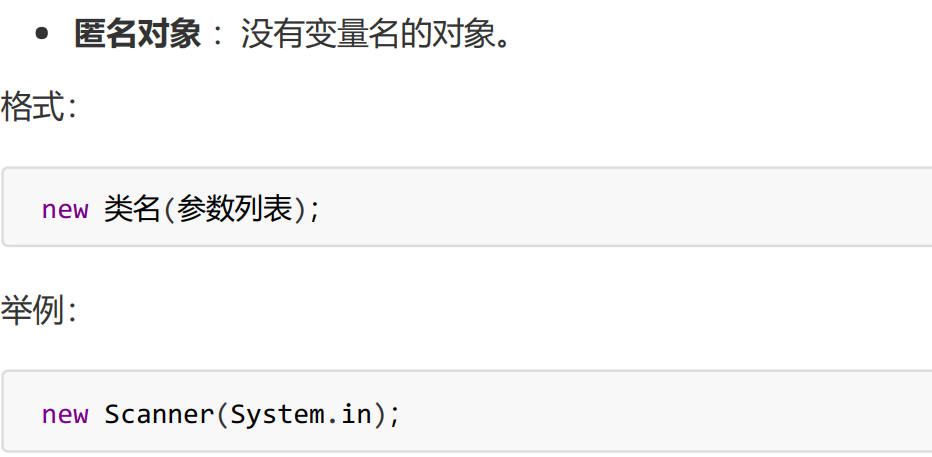
输出结果：绿色表示键盘输入，白色是系统输出





### 匿名对象

使用建议：若确定对象至使用一次，可以使用匿名对象



应用场景

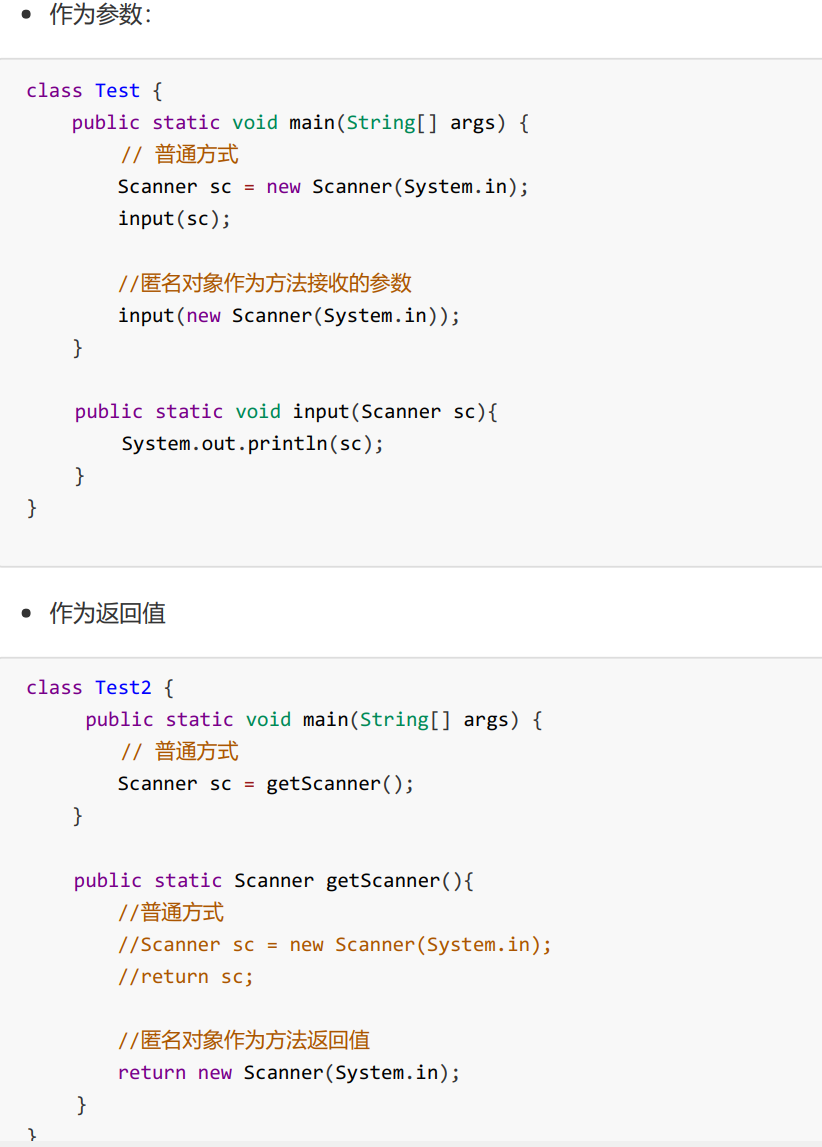
1. 创建匿名对象直接调用方法，没有变量名

new Scanner(System.in).nextInt();

1. 一旦调用两次方法，就是创建了两个对象，造成浪费

小贴士：一个匿名对象，只能使用一次。

1. 匿名对象可以作为方法的参数和返回值



### Random使用



创建一个Random对象，每次调用nextInt（）方法都会生成一个随机数；

例如：r.nextInt(n),取值范围 ：[0,n)

### Array 对象数组

数组内可以存储任何类型数据，包括对象；

缺点：创建之后，程序运行期间长度不可更改；

到目前为止，我们想存储对象数据，选择的容器，只有对象数组。而数组的长度是固定的，无法适应数据变化的需求。为了解决这个问题，Java提供了另一个容器 java.util.ArrayList 集合类,让我们可以更便捷的存储和操作对象数据。

### ArrayList 集合类

* java.util.ArrayList 是大小可变的数组的实现，存储在内的数据称为元素。此类提供一些方法来操作内部存储的元素。 ArrayList 中可不断添加元素，其大小也自动增长。
* 使用步骤：



注意：出现<E>的地方，必须使用某一引用类型代替E，不能用基本类型代替它，以及，确定了一个引用类型，则不可以使用别的引用类型。

对于ArrayList，若不进行赋值直接打印，输出的不是地址值而是空的，会得到 :[]