# static关键字

## 静态变量（类变量）

1. 如果是本类使用，直接就用静态变量名。
2. 如果是其他类使用，可以使用类名来调用，也可以创建一个实例对象来调用。
3. 如果静态变量所在的类是静态类，那么不管在本类里或者在其他外部类，都可以直接使用静态变量名。
4. 静态变量在第一次使用的时候就会初始化，有且只有一个成员对象，可以通过对象调用或者类名.变量名的方式来调用。
5. Java中静态变量**只能在类主体中定义，不能在方法中定义**。 静态变量属于类所有而不属于方法。

## 静态方法（类方法）

1. 可以直接使用类名+方法名的方式来使用静态方法。
2. 在类方法中不能用this来调用本类的类方法。
3. 在类方法中调用本类的类方法时可直接调用。
4. **静态的方法可以被继承，但是不能重写**。如果父类中有一个静态的方法，子类也有一个与其方法名，参数类型，参数个数都一样的方法，并且也有static关键字修饰，那么该子类的方法会把原来继承过来的父类的方法隐藏，而不是重写。
5. 执行一下代码

public class TestClass {

private static void testMethod(){

System.out.println("testMethod");

}

public static void main(String[] args) {

((TestClass)null).testMethod(); //null不是实例对象，可以转换成任意类型

}

}

方法是static静态方法，直接使用"类.方法"即可，因为静态方法使用不依赖对象是否被创建。**null可以被强制类型转换成任意类型（不是任意类型对象）**，于是可以通过它来执行静态方法。

## 静态代码块

1. 无法直接调用静态初始化块。
2. 在创建第一个实例前或引用任何静态成员之前，将自动调用静态初始化块来初始化。
3. 静态初始化块既没有访问修饰符，也没有参数。

# 类的加载

## 1. 父类子类中的加载顺序

1. 父类静态代码块(包括静态初始化块，静态属性，但不包括静态方法)
2. 子类静态代码块(包括静态初始化块，静态属性，但不包括静态方法 )
3. 父类非静态代码块( 包括非静态初始化块，非静态属性 )
4. 父类构造函数
5. 子类非静态代码块 ( 包括非静态初始化块，非静态属性 )
6. 子类构造函数
7. 其中：类中静态块按照声明顺序执行，并且(1)和(2)不需要调用new类实例的时候就执行了(意思就是在类加载到方法区的时候执行的)
8. **构造方法的优先级一般比代码块低。**

静态成员变量或静态代码块>mian方法>非静态成员变量或非静态代码块>构造方法

## 2. 代码例子

public class B {

public static B t1 = new B(); //静态变量

public static B t2 = new B(); //静态变量

{

System.out.println("构造块");

}

static

{

System.out.println("静态块"); //静态块

}

public static void main(String[] args) //静态方法

{

B t = new B();

}

}

1. 静态域最先初始化，静态域包括：静态变量，静态方法，静态块，加载顺序由声明顺序决定。

(2) 静态域：用static申明，JVM加载类时执行，仅执行一次

构造块：类中直接用{}定义，每一次创建对象时执行

加载顺序：静态域>main()>构造块>构造方法

# 选择语句

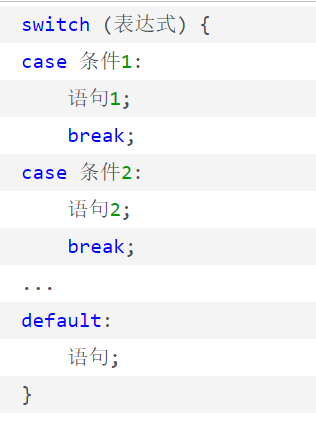
## Switch..Case介绍

1. switch 语句由一个控制表达式和多个 case 标签组成。
2. **switch 控制表达式支持的类型**有

byte、short、char、int、enum（Java 5）、String(Java 7)。

1. switch-case 语句完全可以与 if-else 语句互转，但通常来说，switch-case 语句执行效率要高。
2. default 在当前 switch 找不到匹配的 case 时执行。default 并不是必须的。
3. **一旦 case 匹配，就会顺序执行后面的程序代码，而不管后面的 case 是否匹配，直到遇见 break**。

## 语法格式



# 获取键盘输入Scanner

1. nextInt() 只读取数值，剩下”\n” 还没有读取，并将 cursor 放在本行中
2. next() 方法遇见第一个有效字符（非空格，非换行符）时，开始扫描，当遇见第一个分隔符或结束符 (空格或换行符) 时，结束扫描，获取扫描到的内容，即获得第一个扫描到的不含空格、换行符的单个字符串。
3. nextLine() 时，则可以扫描到一行内容并作为一个字符串而被获取到

