# 代码块

## 三种代码块

### 总览

### 普通代码块

### 构造代码块

### 静态代码块

# 代码块

## 代码块概述

\* 在Java中，使用{}括起来的代码被称为代码块。

## 代码块分类

\* 根据其位置和声明的不同，可以分为局部代码块，构造代码块，静态代码块，同步代码块(多线程讲解)。



## 常见代码块的应用

### 局部代码块—没用

在方法中出现；限定变量生命周期，及早释放，提高内存利用率

### 构造代码块 (初始化块)—没用

在类中方法外出现；多个构造方法方法中相同的代码存放到一起，每次调用构造都执行，并且在构造方法前执行

构造代码块作用：给所有的对象进行统一的初始化。

2：作用

1：给对象进行初始化。对象一建立就运行并且优先于构造函数。

2：与构造函数区别

1：构造代码块和构造函数的区别，构造代码块是给所有对象进行统一初始化， 构造函数给对应的对象初始化。

2：构造代码块的作用：它的作用就是将所有构造方法中公共的信息进行抽取。

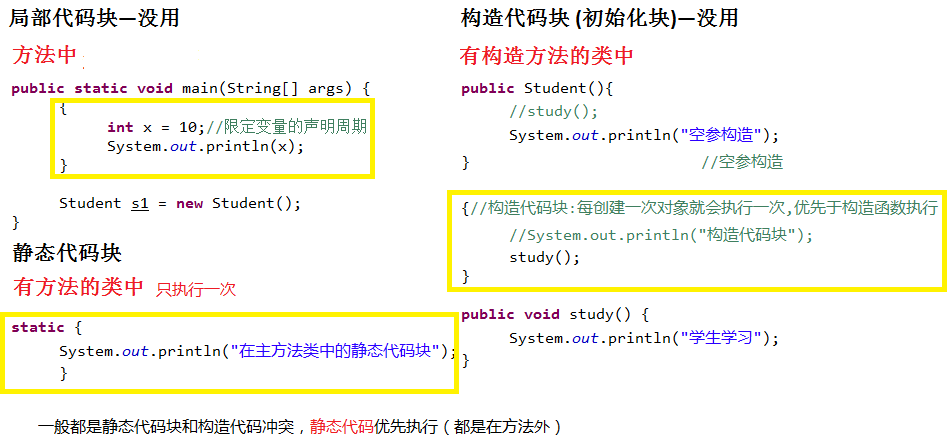
### 静态代码块

1 静态代码块不能存在于任何方法体内。

2 静态代码块不能直接访问静态实例变量和静态实例方法，需要通过类的实例对象来访问。

在类中方法外出现，并加上static修饰；用于给类进行初始化，在加载的时候就执行，并且只执行一次。

一般用于加载驱动



**class** Demo1\_Code {

**public** **static** **void** main(String[] args) {

{

**int** x = 10; //限定变量的声明周期

System.*out*.println(x);

}

Student s1 = **new** Student();

System.*out*.println("---------------");

Student s2 = **new** Student("张三",23);

}

**static** {

System.*out*.println("我是在主方法类中的静态代码块");

}

}

**class** Student {

**private** String name;

**private** **int** age;

**public** Student(){

//study();

System.*out*.println("空参构造");

} //空参构造

**public** Student(String name,**int** age) {//有参构造

//study();

**this**.name = name;

**this**.age = age;

System.*out*.println("有参构造");

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getName() {

**return** name;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** **int** getAge() {

**return** age;

}

{ //构造代码块:每创建一次对象就会执行一次,优先于构造函数执行

//System.out.println("构造代码块");

study();

}

**public** **void** study() {

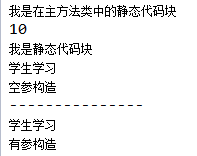
System.*out*.println("学生学习");

}

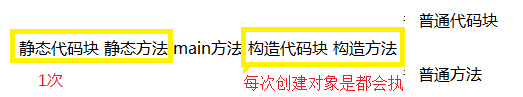
**static** { //随着类加载而加载,且只执行一次

System.*out*.println("我是静态代码块"); //作用:用来给类进行初始化,一般用来加载驱动

} //静态代码块是优先于主方法执行



## 优先级



# 代码块面试题

**class** Student {

**static** {

System.*out*.println("Student 静态代码块");

}

{

System.*out*.println("Student 构造代码块");

}

**public** Student() {

System.*out*.println("Student 构造方法");

}

}

**class** Demo2\_Student {

**static** {

System.*out*.println("Demo2\_Student静态代码块");

}

**public** **static** **void** main(String[] args) {

System.*out*.println("我是main方法");

Student s1 = **new** Student();

Student s2 = **new** Student();

}

}

