34从零开始学Java34之构造方法咋回事?

作者：孙玉昌，昵称【**一一哥**】，另外【**壹壹哥**】也是我哦

千锋教育高级教研员、CSDN博客专家、万粉博主、阿里云专家博主、掘金优质作者

前言

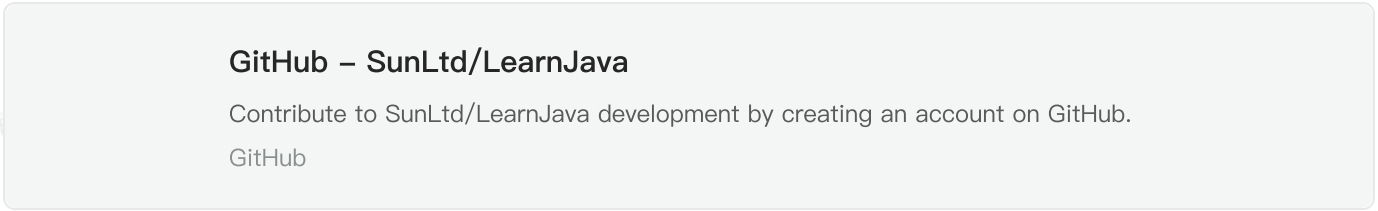
在前面的几篇文章中，**壹哥**给大家介绍了不少关于方法的内容，这些内容是我们日常开发时的必备基础，所以你必须要牢牢掌握哦。接下来在今天的文章中，**壹哥**会给大家介绍关于方法的另一个重磅知识--构造方法！话说构造方法也是方法，那么构造方法该怎么定义？有什么特殊之处？与普通方法的区别何在？要想知道这些，就请大家跟着**壹哥**来学习一下吧。

----------前戏已做完，精彩即开始----------

全文大约【4100】字，不说废话，只讲可以让你学到技术、明白原理的纯干货！本文带有丰富的案例及配图视频，让你更好地理解和运用文中的技术概念，并可以给你带来具有足够启迪的思考.....

配套开源项目资料

Github：



Gitee：

一. 构造方法简介

1. 概念

首先我们来了解一下到底啥是**构造方法(也被称为构造器)**。

不管咋样，构造方法也是方法，所以普通方法具有的大多数特点，构造方法基本上也都拥有！**壹哥**这么一说，构造方法和普通方法，好像还不完全一样？是的！构造方法有其特殊之处！

我们来看名字，”构造“，啥意思？通俗地解释，就是”创建、建造“的意思，所以**构造方法的最主要用途就是来初始化类对象的！也就是用来创建新对象的！**而构造方法的特殊之处，主要表现在两

个方面：

> 1. 可以创建类对象；
> 2. 构造方法没有返回值。

以上这两个基本特征，都是普通方法所不具备的！

2. 作用

根据构造方法的概念可知，**构造方法的核心作用就在于初始化类的新对象。**一般情况下，我们是利用new运算符结合构造方法，创建出一个新对象。并且**在构造方法中，我们还可以给一个类的实例变量赋初值，或者执行其它必要的步骤来创建出一个完整的对象。**

有的小伙伴说，我们在之前的案例中，好像并没有自己定义过构造方法啊？那怎么也创建出了对象呢？

实际上，**在Java的每个类中，都会有一个默认的无参构造方法！**也就是说，只要我们创建了一个Java类，不管你自己有没有定义构造方法，Java都会自动给你创建一个不带参数的构造方法，只是默认情况下我们没有看到罢了。**默认的构造方法，其访问修饰符和类的访问修饰符相同。**比如类是public修饰的，默认的构造方法也是public修饰的；类是protected修饰的，默认的构造方法也是protected修饰的，以此类推。

但大家要注意，**一旦你自己定义了构造方法，编译器就*不再*自动给我们创建默认的构造方法了！**

本节配套视频如下：

https://player.bilibili.com/player.html?bvid=BV1Ja411x7XB&p=74&page=74

3. 特点(重点)

因为构造方法的特殊作用，所以具有与众不同的一些特点：

> - 构造方法的名称必须与类名相同；
> - 构造方法可以有0个、1 个或多个参数；
> - 构造方法没有任何返回值，也不带void关键字；
> - 构造方法不能被static、final、synchronized、abstract 和native修饰；
> - 构造方法的默认返回类型就是对象类型本身；
> - 构造方法只能与new运算符结合使用；
> - 一个类中可以有多个同名的构造方法，但参数不同，即构造方法也可以被重载；

> - 所有的类都有一个默认的构造方法。

对于以上这些特点，**壹哥**希望大家能够牢牢的记住。

二. 基本使用

1. 基本语法

在一个类中，与类名相同的方法就是构造方法。每个类中可以有多个构造方法，但要求它们的参数不同。构造方法的基本语法如下所示：

Java | 复制代码

```java
public class 类名{

    //构造方法名与类名相同!
    public 类名(){
        ......
    }

    //带参数的构造方法
    public 类名(参数列表...){
        ......
    }

}

```

从上面的语法中我们可以看出，构造方法好像是没有返回值的，既没有声明返回值类型，也没有void关键字。但实际上，**如果我们给构造方法设置了返回值类型或者添加了void关键字，该方法在编译时也不会出错，因为Java会把这个方法当成普通的方法来处理。**

虽然在表面上看，构造方法没有返回值，也不用我们带void关键字，**但其实类的构造方法是有返回值的！**当我们**使用new关键字来调用构造方法时，构造方法会返回该类的一个实例，这个实例其实就是构造方法的返回值。**因为构造方法的返回值类型总是当前类，所以不用我们明确定义返回值类型。但**我们不能在构造方法里明确使用return来返回当前类对象，因为构造方法的返回值是隐式的。**

2. 基本案例

接下来**壹哥**就给大家设计一个案例，来给大家讲解构造方法的基本使用。

Java | 复制代码

```java
/**
 * @author ——哥Sun
 * QQ: 2312119590
 * CSDN、掘金、知乎找我哦
 */
public class Student {

    private String name;

    private int age;

    private String sex;

    private double score;

    public Student() {
        // 当我们自己显式地创建了构造方法后，默认的构造方法即失效
        System.out.println("这是无参的构造方法");
    }

    //构造方法重载
    public Student(String name) {
        System.out.println("这是有参的构造方法");
        // 我们可以在构造方法中，进行类的初始化操作
        System.out.println("name=" + name);
    }

    //构造方法重载
    public Student(String name, String sex, double score) {
        System.out.println("这是有参的构造方法");
        // 我们可以在构造方法中，进行类的初始化操作
        this.name = name;
        //直接在构造方法中给变量赋初值
        this.sex = "男";
        this.score = 99;
    }

}
```

类的构造方法不是必须要定义的。如果我们在类中没有定义构造方法，Java会自动为该类生成一个默认的构造方法，默认的构造方法不包含任何参数，并且方法体为空。

在这个案例中，**壹哥**定义了3个构造方法，无参和有参的构造方法都有。当我们定义了多个构造方法，在通过new操作符调用时，**编译器会根据我们传递的参数数量、位置和类型自动区分该调用哪个构造方法**。在一个类中同时定义多个同名的方法，这就是方法的重载，后面我会再细讲！

大家要注意，**无参数的构造方法也被称为Nullary构造方法**。只有程序自带的构造方法，才称为默认构造函数。如果是**我们自己编写的无参、没有内容的构造方法，并不称是默认构造函数了，而是叫做Nullary构造方法**。

本节配套视频链接如下：

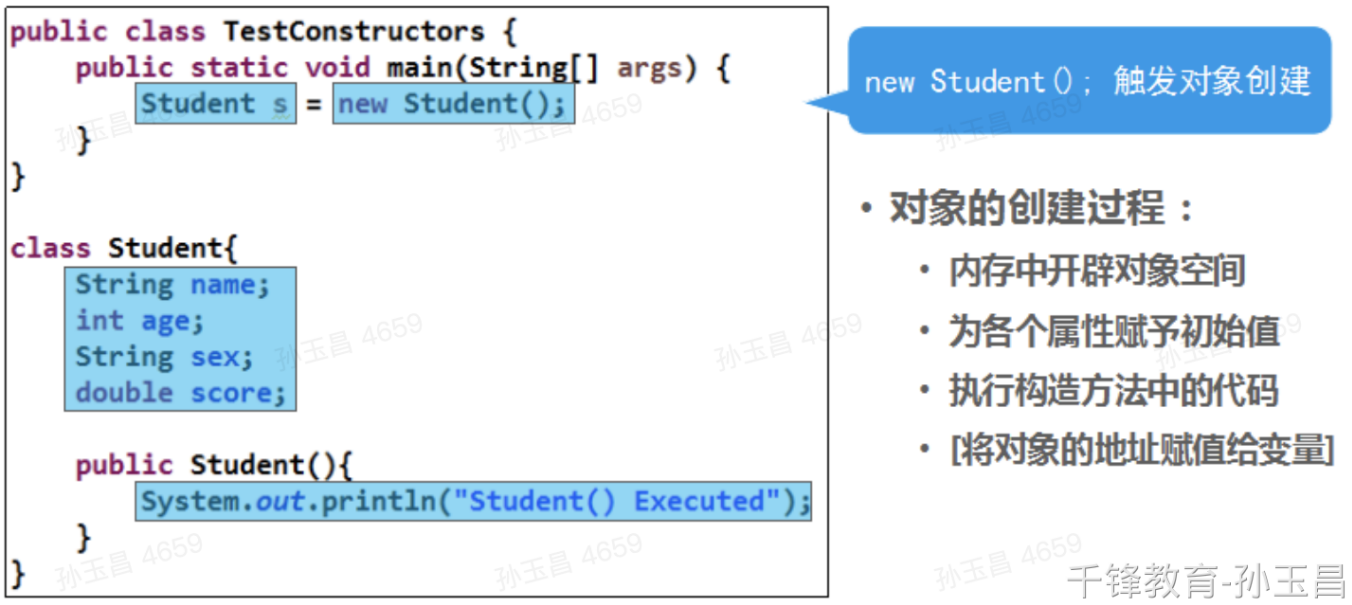
https://player.bilibili.com/player.html?bvid=BV1Ja411x7XB&p=75&page=75

this关键字

在上述案例代码中，我们用到了一个this关键字。**this表示是当前类的对象，通俗地说就是表示当前类对象”自己“，是在对象被创建时自动产生的**。我们**使用this，用来调用本类的属性、方法、构造方法。**当我们在构造方法中使用this时，this表示是当前类的成员变量。关于this关键字，**壹哥**会在后面专门利用一篇文章进行讲解，此处我们先暂时了解这么多即可。

3. 对象创建过程分析

我们根据Student类对象的创建，来分析一下Java对象的创建过程，如下图所示：

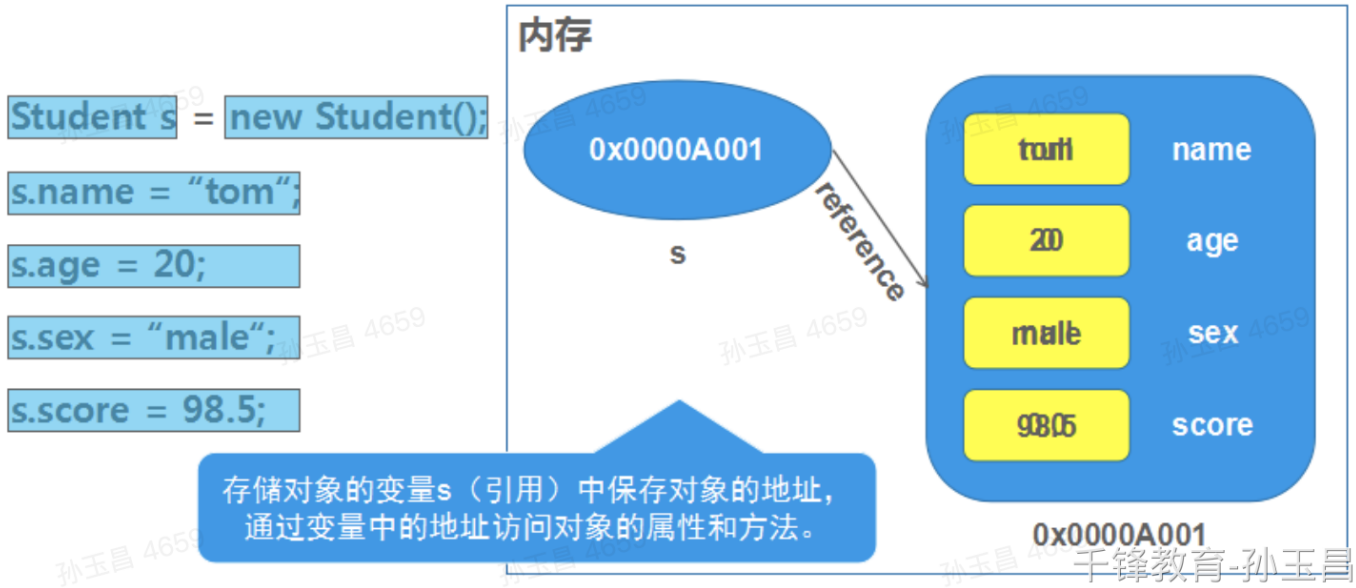


根据上图可知，一个对象的创建，其核心过程如下：

1. 内存中开辟一个对象空间；

> 2. 为对象的各个属性赋初始值；
> 3. 执行构造方法中的代码；
> 4. 将对象的内存地址赋值给Java变量。

从上面的过程可知，对象创建之后，对象的内存地址会赋值给变量s，我们可以通过变量s中的地址来访问对象的属性和方法，如下图所示：



三. 知识拓展

1. 构造方法互调

当我们定义了多个构造方法时，一个构造方法可以调用其他的构造方法，这样便于进行代码复用。在一个构造方法中调用其他构造方法的基本语法是使用this(...)，并且this()操作要放在构造方法的第一行，否则编译时就会有问题出现：Constructor call must be the first statement in a constructor。

Java | 复制代码

```java
/**
 * @author ——哥Sun
 * QQ: 2312119590
 * CSDN、掘金、知乎找我哦
 */
public class Teacher {

    private String name;

    private int age;

    private double salary;

    public Teacher() {
        // 当我们自己显式地创建了构造方法后，默认的构造方法即失效
        //通过this调用另一个构造方法Teacher(String)
        this("——哥");
    }

    // 构造方法重载
    public Teacher(String name, int age, double salary) {
        // 我们可以在构造方法中，进行类的初始化操作
        this.name = name;
        this.age = age;
        this.salary = salary;
    }

    // 构造方法重载
    public Teacher(String name) {
        //在一个构造方法中，通过this调用另一个构造方法Teacher(String, int,double)。
        //大家要注意，当我们在构造方法中调用另一个构造方法时，this()操作必须放在构造方法的第一行，注释语句除外!
        //否则会产生如下异常:
        //Constructor call must be the first statement in a constructor
        this(name, 28, 18888.88);
    }

}
```

2. 构造方法私有化

2.1 几个不能用的修饰符

壹哥在前面说过，**构造方法不能被static、final、synchronized、abstract和native 这几个关键字修饰**。因为构造方法用于初始化类的新对象，使用static修饰没有意义。构造方法不能被子类继承，所以使用 final和abstract修饰也没有意义。多个线程不会同时创建出内存地址相同的同一个对象，所以用 synchronized修饰没有必要。当然，对于以上几个关键字的详细作用，我们现在还不够了解，**壹哥**后面会针对这些关键字进行详细讲解，请耐心等待哦。

2.2 单例模式简介

除了以上这几个关键字之外，其他的很多关键字，构造方法都是可以使用的。比如构造方法的访问修饰符，一般都是public，但在某些特殊的场景下，也可以使用protected或private来修饰构造方法。当我们使用private来修饰构造方法时，这就是所谓的构造方法私有化，主要是在单例模式中使用。

单例模式是23种设计模式中创建型模式的一种。通过单例模式创建的类对象，可以保证在当前进程或者线程中只有一个实例。单例模式有两种比较常见的实现方式：**饿汉式**、**懒汉式**。

关于单例模式，以后**壹哥**这边会有专门的设计模式专栏，敬请期待哦，此处先简单介绍一下。

2.3 代码实现

在下面的代码中，壹哥设计了一个Wife妻子类。我们知道，不管你有过多少段婚姻，你合法的妻子最多只会有一个，不可能有多个，那么如何保证你只有一个妻子对象呢(太多了怕你受不了)？请看下面的代码：

Java | 复制代码

```java
/**
 * @author ——哥Sun
 * QQ：2312119590
 * CSDN、掘金、知乎找我哦
 */
public class Wife {

    // 创建一个私有静态的实例对象
    private static Wife wife = new Wife();

    //私有化构造方法
    private Wife() {
    }

    // 创建一个供外界获取当前实例对象的公开静态方法
    public static Wife getInstance() {
        return wife;
    }

    //测试对象的创建，也可以在另外一个类中调用
    public static void main(String[] args) {
        // 获取Wife对象1
        Wife wife1 = Wife.getInstance();
        // 获取Wife对象2
        Wife wife2 = Wife.getInstance();

        // 打印结果会是 true，说明两个对象是同一个对象
        System.out.println(wife1 == wife2);
    }

}
```

上面的代码，我们是基于”饿汉式“来实现的单例模式。在构造方法被私有化后，我们只能通过它提供的公开方法 getInstance()，来获取当前类的实例对象。而我们每次获取到的对象都是同一个，保证了Wife对戏的唯一性。

而从 Wife 类中，我们能看得到它有个静态的私有属性，这个私有属性就是当前类的实例对象。也就是说，不管我们有没有调用这个类，类中的静态对象属性在类加载进内存的时候就已经存在了，这就是饿汉式的特点。

饿汉式就婚姻里的对象一样，他(她)一早就出现在这个世上了，只是需要我们彼此寻找到对方。每个单身已久的男人，都会如同”饿汉“一样，饥不可耐的想要品尝Wife的美妙，嘻嘻。

----------正片已结束，来根事后烟----------

四. 结语

至此，**壹哥**就把构造方法的使用及其特点给大家介绍完毕了，现在你学会了吗？接下来**壹哥**给大家总结一下本文的重点：

> - 我们在创建对象实例时，会通过new操作符调用对应的构造方法；
> - 构造方法用于对象的创建和初始化；
> - 如果我们没有定义构造方法，编译器会自动创建一个默认的无参数构造方法；
> - 一个类中可以定义多个构造方法，编译器会根据参数自动判断调用哪个；
> - 可以在一个构造方法中调用另一个构造方法，便于代码复用；
> - 构造方法可以私有化，以此可以实现单例模式。

另外如果你独自学习觉得有很多困难，可以加入**壹哥**的学习互助群，大家一起交流学习。

五. 配套视频

如果你不习惯阅读技术文章，或是对文中的技术概念不能很好地理解，可以来看看**壹哥**帮你筛选出的视频教程。**与本文配套的Java学习视频，链接如下：**

https://player.bilibili.com/player.html?bvid=BV1FK4y1x7Ny&p=42&page=42

六. 今日作业

1. 第一题

设计一个汽车类，定义几个构造方法，并在构造方法中对其属性赋值，然后创建出汽车类对象。