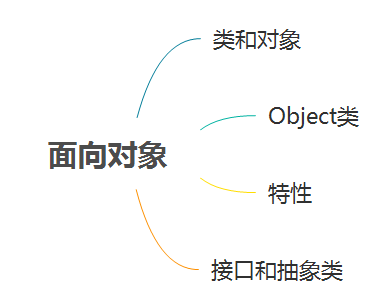
# 面向对象



## 类和对象

### 什么是类,对象

类是一个模板,描述它的对象的属性和行为

对象是类的实例

### 访问修饰符

　　　　  访问权限   类   包  子类  其他包

  　　　　  public     ∨   ∨    ∨     ∨          （对任何人都是可用的）

   　　　　 protect    ∨   ∨   ∨     ×　　　 （继承的类可以访问以及和private一样的权限）

   　　　　 default    ∨   ∨   ×     ×　　　 （包访问权限，即在整个包内均可被访问）

   　　　　 private    ∨   ×   ×     ×　　　 （除类型创建者和类型的内部方法之外的任何人都不能访问的元素）

### 构造方法

默认构造

重载构造

## Object类

### 什么是Object类

Java中所有类的父类

### Object的常用方法

equals()

判断两个对象是否相等

toString()

返回对象的字符串表示形式

hasCode()

返回对象的哈希吗值,这个方法是为了更好的实现哈希表

wait()

使当前线程挂起,并放弃CPU使用

notify()

唤醒某个被执行wait()方法线程

notifyAll()

唤醒所有被执行wait()方法的线程

## 面向对象特性

### 什么是java面向对象特性

三大特性,封装,多态,J继承

### 什么是封装

将部分数据和方法封装起来,只能通过指定方法调用

#### 为什么使用封装

屏蔽方法的复杂性,提高代码的健壮性

### 什么是多态

同一种行为的不同表现形式

#### 多态的实现

重写

接口

抽象类和抽象方法

### 什么是继承

子类获得父类的部分属性和方法

#### 为什么使用继承

减少代码量

#### 继承的特性

子类拥有父类非 private 的属性、方法。

子类可以拥有自己的属性和方法，即子类可以对父类进行扩展。

子类可以用自己的方式实现父类的方法。

子类通过super关键字访问父类

Final类无法被继承

子类与父类的同名方法,父类方法被覆盖,同名变量就是遮盖,不会覆盖父类变量

Java 的继承是单继承，但是可以多重继承，单继承就是一个子类只能继承一个父类，多重继承就是，例如 A 类继承 B 类，B 类继承 C 类，所以按照关系就是 C 类是 B 类的父类，B 类是 A 类的父类，这是 Java 继承区别于 C++ 继承的一个特性。

提高了类之间的耦合性（继承的缺点，耦合度高就会造成代码之间的联系越紧密，代码独立性越差）。

## 接口和抽象类

### 什么是接口?什么是抽象类?

抽象方法的合集是接口,不能实例化对象的类是抽象类

### 为什么使用接口和抽象类

接口可以解耦,接口可以实现多继承的思想

### 接口和抽象类的区别

|  |  |  |
| --- | --- | --- |
| **参数** | **抽象类** | **接口** |
| 默认的方法实现 | 它可以有默认的方法实现 | 接口完全是抽象的。它根本不存在方法的实现 |
| 实现 | 子类使用**extends**关键字来继承抽象类。如果子类不是抽象类的话，它需要提供抽象类中所有声明的方法的实现。 | 子类使用关键字**implements**来实现接口。它需要提供接口中所有声明的方法的实现 |
| 构造器 | 抽象类可以有构造器 | 接口不能有构造器 |
| 与正常Java类的区别 | 除了你不能实例化抽象类之外，它和普通Java类没有任何区别 | 接口是完全不同的类型 |
| 访问修饰符 | 抽象方法可以有**public**、**protected**和**default**这些修饰符 | 接口方法默认修饰符是**public**。你不可以使用其它修饰符。 |
| main方法 | 抽象方法可以有main方法并且我们可以运行它 | 接口没有main方法，因此我们不能运行它。 |
| 多继承 | 抽象方法可以继承一个类和实现多个接口 | 接口只可以继承一个或多个其它接口 |
| 速度 | 它比接口速度要快 | 接口是稍微有点慢的，因为它需要时间去寻找在类中实现的方法。 |
| 添加新方法 | 如果你往抽象类中添加新的方法，你可以给它提供默认的实现。因此你不需要改变你现在的代码。 | 如果你往接口中添加方法，那么你必须改变实现该接口的类。 |

### 题目

#### 2、Java有哪些特性，举个多态的例子。

java的三大特性:封装,继承和多态

多态(多态是同一个行为具有多个不同表现形式或形态的能力。)，

子类重写父类中的可访问的方法

#### 3.String为啥不可变

因为String对象中保存字符串的是一个名为value的字符数组,它被private final修饰,所以String一旦初始化就不会改变。

(知识拓展:只所以String可以被重新赋值,是因为新建了一个String而不是被赋值)

#### 4.类和对象的区别

类是一种类型,对象是类的实例

#### 5.请列举你所知道的Object类的方法

equals(),toString(),hashCode(),getClass(),notify(),wait()

#### 6、重载和重写的区别？相同参数不同返回值能重载吗？

1)参数列表:重载必须修改,重写不能修改

返回类型:重载可以修改,重写不能修改

修饰符:重载可以修改,重写不能更严格的限制

2)不能

#### 7、”static”关键字是什么意思？Java中是否可以覆盖(override)一个private或者是static的方法？

1)静态的意思

2)不能

#### 8.String能继承吗？

不能

String的完整定义：public final class String extends Object

#### 11.静态变量存在哪?

全局区（静态区）)

#### 12.讲讲什么是泛型

泛型就是容器类型,用来构建某一类型的容器。

(知识拓展:

javac中可以看到被泛型修饰后的类型

Jvm中只能看到原始类型)

#### 13、解释extends 和super 泛型限定符-上界不存下界不取

上界 <? extend Fruit>,表示所有继承Fruit的子类,,但是具体是哪一种子类无法确定,所以add的时候不知道具体是哪一种类型,但是get的时候可以向上转型成父类Fruit所以可以用get无法用add

下界 <? super Apple>,表示Apple的所有父类,包括Fruit,add的时候不能添加Fruit的父类,因为无法确定到底是哪一个父类,但是可以添加Apple的子类,因为Apple的子类都可以向上转型成Apple或Apple的父类,但是get的时候Apple的父类无法确定

总结就是编译器支持向上转型不支持向下转型

#### 14.是否可以在static环境中访问非static变量

不可以，因为static变量是属于类的，在类加载的时候就被初始化了，这时候非静态变量并没有加载

就是（static变量在类加载的时候就有了,而此时对象还并没有初始化,只有要new的时候对象才有）

#### 16.Java支持多继承么？

类不支持,接口支持

#### 17.接口和抽象类的区别

|  |  |  |
| --- | --- | --- |
| **参数** | **抽象类** | **接口** |
| 默认的方法实现 | 它可以有默认的方法实现 | 接口完全是抽象的。它根本不存在方法的实现 |
| 实现 | 子类使用**extends**关键字来继承抽象类。如果子类不是抽象类的话，它需要提供抽象类中所有声明的方法的实现。 | 子类使用关键字**implements**来实现接口。它需要提供接口中所有声明的方法的实现 |
| 构造器 | 抽象类可以有构造器 | 接口不能有构造器 |
| 与正常Java类的区别 | 除了你不能实例化抽象类之外，它和普通Java类没有任何区别 | 接口是完全不同的类型 |
| 访问修饰符 | 抽象方法可以有**public**、**protected**和**default**这些修饰符 | 接口方法默认修饰符是**public**。你不可以使用其它修饰符。 |
| main方法 | 抽象方法可以有main方法并且我们可以运行它 | 接口没有main方法，因此我们不能运行它。 |
| 多继承 | 抽象方法可以继承一个类和实现多个接口 | 接口只可以继承一个或多个其它接口 |
| 速度 | 它比接口速度要快 | 接口是稍微有点慢的，因为它需要时间去寻找在类中实现的方法。 |
| 添加新方法 | 如果你往抽象类中添加新的方法，你可以给它提供默认的实现。因此你不需要改变你现在的代码。 | 如果你往接口中添加方法，那么你必须改变实现该接口的类。 |

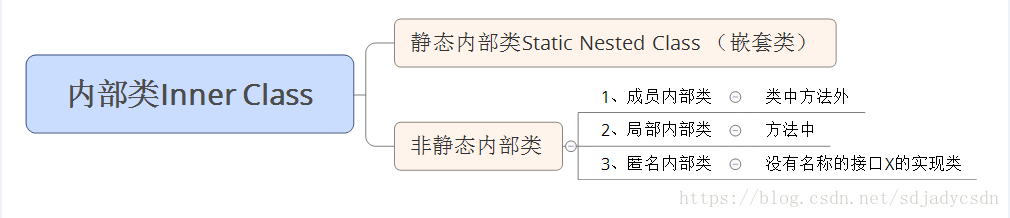
#### 20、final, finally, finalize的区别。

fianl是一个修饰符可以用来修饰类,变量和方法

finally是java中的一个异常处理机制,finally只能用在try/catch语句中并且附带着一个语句块,总会执行,除开之前有（System.out.exit（0）情况）

finallze是object类中的一个方法,在GC准备释放对象所占用的内存空间之前，它将首先调用finalize()方法。

#### 23、Static Nested Class 和 Inner Class的不同



#### 24、当一个对象被当作参数传递到一个方法后，此方法可改变这个对象的属性，并可返回变化后的结果，那么这里到底是值传递还是引用传递?

java只有值传递

因为此时参数在内存的栈中的值是这个对象的地址

（无论是基本类型作为参数传递，还是对象作为参数传递，实际上传递的都是值，只是值的的形式不用而已.

都是传递的栈内存中的值,只不过基本类型在栈内存里面存的就是值,而引用类型里面在栈内存中存的就是堆内存中的地址,所以对象当参数的时候,就会新建一个对象副本,这个副本在栈内存中保存的也是堆内存的地址。）

#### 27、内部类可以引用他包含类的成员吗？有没有什么限制？

非静态内部类没有限制

静态内部类只能引用类的静态成员

#### 28、两个对象值相同(x.equals(y) == true)，但却可有不同的hash code说法是否正确？

不正确,两个对象相同一定会有相同的哈希码,反之却不一定成立

#### 29、重载（Overload）和重写（Override）的区别。重载的方法能否根据返回类型进行区分？

1）重载发生在同一个类中,相同方法名不同的参数列表即为重载。重写发生于子类与父类之间,子类覆盖父类中的某一个方法(除开构造方法,final方法),参数列表和返回类型必须一致，访问权限不能比父类中被重写的方法的访问权限更低

2）因为调用的时候编译器不知道类型信息（例如 float max(int a,int b

int man(int a,int b)

max(1,2)编译器不知道调用哪一个）

**32、**

#### 34、Java中，什么是构造函数？什么是构造函数重载？什么是复制构造函数？

构造函数就是对象被创建的时候调用的函数,构造函数重载就是可以有多个不同参数列表的构造函数。java没有复制构造函数

#### 35、hashCode()和equals()方法有什么联系？

hasCode方法和equals方法的联系只有在set集合和map集合中体现

如果集合中新添元素,先用hasCode方法判断是否集合类有和新元素相等的哈希码,如果没有就添加新元素,有相同的哈希码再比较equals方法,不相等就加入。