**修饰符**

Sprivate和prtected不能修饰类和接口。接口里的变量都隐式声明为public static final,而接口里的方法默认情况下访问权限为public，所以和有没有default没多大关系。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **修饰符** | **当前类** | **同一包内** | **子孙类** | **其他包** |
| public | Y | Y | Y | Y |
| protected | Y | Y | Y | N |
| default | Y | Y | N | N |
| private | Y | N | N | N |

**Static修饰符：**

Static修饰的方法和变量是属于该类的，可通过该类的实例化对象进行直接调用，局部变量不能使用static修饰，静态方法不能使用类的非静态变量。

**Final修饰符：**

final 变量能被显式地初始化并且只能初始化一次。被声明为 final 的对象的引用不能指向不同的对象。但是 final 对象里的数据可以被改变。也就是说 final 对象的引用不能改变，但是里面的值可以改变。

Final修饰的方法能被继承，但是不能修改。（修饰方法的目的就是为了内容被修改）

Final修饰的类不能被继承。

**abstract修饰符：**

抽象类不能被实例化，抽象类的唯一目的就是为了方便以后的扩充。

一个类不能同时被abstract和final修饰，一个类中如果有被abstract修饰的方法， 那么该类也必须被abstract修饰。

抽象方法是一种没有任何实现的方法，具体交由子类去实现。

抽象方法不能被static和final修饰。

抽象方法以分号结尾，如**public abstract sample(); 。**

继承抽象类的子类，必须实现父类的所有抽象方法，除非它也是抽象类。

如果一个类中有若干个抽象方法，那么该类也必须被抽象化。抽象类中不一定有抽象方法。

**synchronized 修饰符**

synchronized 关键字声明的方法同一时间只能被一个线程访问。synchronized 修饰符可以应用于四个访问修饰符。

实例

public synchronized void showDetails(){.......}

**transient 修饰符**

序列化的对象包含被 transient 修饰的实例变量时，java 虚拟机(JVM)跳过该特定的变量。

该修饰符包含在定义变量的语句中，用来预处理类和变量的数据类型。

实例

public transient int limit = 55;   // 不会持久化

public int b; // 持久化

**volatile 修饰符**

volatile 修饰的成员变量在每次被线程访问时，都强制从共享内存中重新读取该成员变量的值。而且，当成员变量发生变化时，会强制线程将变化值回写到共享内存。这样在任何时刻，两个不同的线程总是看到某个成员变量的同一个值。