抽象类abstract

抽象类是用来捕捉子类的通用特性的 。它不能被实例化，只能被用作子类的超类。抽象类是被用来创建继承层级里子类的模板。

抽象类继承默认继承其所有方法（不必写出来），可以直接重写某个方法，也可以新加方法。若方法中含有父类某些成员变量，也不必写出，默认继承含有父类成员变量。new对象时 ， 具备左侧的所有方法（右侧新加方法不具备），具体实现按右侧执行。

New 对象时可以直接新建父类抽象类，可以在new后 默认继承父类方法，可重写，但不能添加方法。

接口Interface

接口是抽象方法的集合。如果一个类实现了某个接口，那么它就继承了这个接口的抽象方法。这就像契约模式，如果实现了这个接口，那么就必须确保使用这些方法。接口只是一种形式，接口自身不能做任何事情。

继承一个接口时

当你实现这个接口时，你就需要实现接口的所有方法（必须写出来） ， new对象时，具备左侧的所有方法和变量，方法实现按右侧实现，变量值按左侧实现（且只具备左侧的变量），但 不具备右侧新加的方法。

继承接口的变量任何类型都可以改变值（包括final ， static ， static final）

当继承多个接口时，变量重复时，不继承（可以在实现类中重写定义），方法重复时，只需要实现一次就可以。

New对象时（接口 a = new 实现类 实现类继承多个接口） 变量按左侧接口，具体实现方法 按右侧实现类，左侧接口中不具备的方法 即使右侧具备，也不能使用

什么时候使用抽象类和接口

如果你拥有一些方法并且想让它们中的一些有默认实现，那么使用抽象类吧。

如果你想实现多重继承，那么你必须使用接口。由于Java不支持多继承，子类不能够继承多个类，但可以实现多个接口。因此你就可以使用接口来解决它。

如果基本功能在不断改变，那么就需要使用抽象类。如果不断改变基本功能并且使用接口，那么就需要改变所有实现了该接口的类。



Java8中的默认方法和静态方法

Oracle已经开始尝试向接口中引入默认方法和静态方法，以此来减少抽象类和接口之间的差异。现在，我们可以为接口提供默认实现的方法了并且不用强制子类来实现它。