到底什么是多态呢？

**官方说：**

* 接口的多种不同的实现方式即为多态。
* 多态性是允许你将父对象设置成为一个或更多的他的子对象相等的技术。
* 我们在程序中定义的引用变量所指向的具体类型和通过该引用变量的方法调用在编程的时候并不确定，当处于运行期间才确定。就是这个引用变量究竟指向哪一个实例对象，在编译期间是不确定的，只有运行期才能确定，这样不用修改源码就可以把变量绑定到不同的类实例上，让程序拥有了多个运行状态，这就是多态。

**说人话：**

允许将子类类型的指针赋值给父类类型的指针，把不同的子类对象都当作父类来看。比如你家有亲属结婚了，让你们家派个人来参加婚礼，邀请函写的是让你爸来，但是实际上你去了，或者你妹妹去了，这都是可以的，因为你们代表的是你爸，但是在你们去之前他们也不知道谁会去，只知道是你们家的人。可能是你爸爸，可能是你们家的其他人代表你爸参加。这就是多态。

多态又分为 编译时多态和运行时多态。  
编译时多态：比如重载  
运行时多态：比如重写

**多态的实现机制**

简单版本：

原理也很简单，父类或者接口定义的引用变量可以指向子类或者具体实现类的实例对象，由于程序调用方法是在运行期才动态绑定的，那么引用变量所指向的具体实例对象在运行期才确定。所以这个对象的方法是运行期正在内存运行的这个对象的方法而不是引用变量的类型中定义的方法。

术语版本：

我们将引入Java静态分派和动态分派这个概念。

* 静态分派:所有依赖静态类型来定位方法执行版本的分派动作。静态分派发生在编译阶段，因此确定静态分派的动作实际上不是由虚拟机来执行的，而是由编译器来完成。（编译时多态）
* 动态分派：在运行期根据实际类型确定方法执行版本的分派动作。（运行时多态）