链滴

# abstract 抽象类的释义

作者：sologxl

原文链接：https://ld246.com/article/1541649749551

来源网站：链滴

抽象类说准确一点，其实也是一个开发模式，开发思维。不算一个技术。

比如说人会跑，车也会跑。

那这样就可以把“跑”这件事抽象出来，虽然实际上的实现逻辑不一样，但是那是实现者的事情。

比如我们可以这样定义：

```
public abstract class sport{
public abstract void run ();
}
```

然后人类要继承这个抽象行为：

```
public class People extends Sport{
@Overrion
public void run(){
System.out.pritnln(" peopel run ");
}
}
```

然后车子也要继承这个抽象行为：

```
public class Car extends Sport{
@Overrion
public void run(){
System.out.pritnln(" car run ");
}
}
```

简单的一比是不是?

另外做个小笔记：

1、抽象类是有构造方法的（当然如果我们不写，编译器会自动默认一个无参构造方法）。而且从结来看，和普通的继承类一样，在new 一个子类对象时会优先调用父类（这里指的是抽象类Car）的构器初始化，然后再调用子类的构造器。至此相信大家都会有这样一个疑问，为什么抽象方法不能实例却有构造器呢？ 对于这个问题网上也中说纷纭，没有确定答案。

我是这样想的：既然它也属于继承的范畴，那么当子类创建对象时必然要优先初始化父类的属性变量实例方法，不然子类怎么继承和调用呢？而它本身不能实例化，因为它本身就是不确定的一个对象，果它能被实例化，那么我们通过它的对象来调用它本身的抽象方法是不是有问题。所以不能实例化有情理之中。因此大家只要记住这个规定就行。