

抽象类和抽象方法

1. 抽象类

- (1) 在正常定义类的面前加上abstract关键字，构成抽象类。不能实例化它
如abstract class A{}
- (2) 作用：规范一些子类的共同特性，自己不做具体描述(实现)
- (3) 规范方式：继承它
也就是说，抽象类使命是作为基类，子类继承它的特性，即起了规范作用

2. 抽象方法

- (1) 是一个无方法体，也无大括号的方法定义头而已，前面需加上abstract。
如abstract function f1(\$x,\$y,\$m);
- (2) 作用：配合抽象类进行对下级类进行行为规范。即相当于要求下级去具体完成一些自己规定的任务

3. 注意

- (1) 一个方法定义为抽象方法，则所在类必须声明为抽象类
- (2) 但抽象类可以没有抽象方法，但这通常意义不大
- (3) 实际利用了继承的重写

eg: 定义抽象的怪物类。有攻击，防守方法

定义一个恐龙类继承自怪物类。重写攻击，防守方法，实现其独有特效