1. Java中访问控制

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修饰词 | 本类内部 | 同一个包内 | 子类 | 其他类 |
| Private | 是 | 否 | 否 | 否 |
| Protected | 是 | 是 | 是 | 否 |
| Public | 是 | 是 | 是 | 是 |
| default | 是 | 是 | 否 | 否 |

1. PHP中访问控制权限

|  |  |  |  |
| --- | --- | --- | --- |
| 修饰词 | 本类内部 | 子类 | 其他类 |
| Private | 是 | 否 | 否 |
| Protected | 是 | 是 | 否 |
| public | 是 | 是 | 是 |

1. PHP中数字是有符号类型的，最大支持2的64次方。
2. 接口和抽象类的区别

首先接口和抽象类的设计目的就是不一样的。**接口是对动作的抽象，而抽象类是对根源的抽象**。

**抽象类**：比如男人，女人这两个类，那我们可以为这两个类设计一个更高级别的抽象类--人。

**接口**：我们可以坐着吃饭，可以站着吃饭，可以用筷子吃饭，可以用叉子吃饭，甚至可以学三哥一样用手抓着吃饭，那么可以把这些吃饭的动作抽象成一个接口--吃饭。

所以在高级语言中（如Java,C#），一个类只能继承一个抽象类（因为你不可能同时是生物又是非生物）。但是一个类可以同时实现多个接口，比如开车接口，滑冰接口，踢足球接口，游泳接口。

抽象类和接口的对比

**1、抽象类和接口都不能被直接实例化，如果二者要实例化，就涉及到多态**。

如果抽象类要实例化，那么抽象类定义的变量必须指向一个子类对象，这个子类继承了这个抽象类并实现了这个抽象类的所有抽象方法。

如果接口要实例化，那么这个接口定义的变量要指向一个子类对象，这个子类必须实现了这个接口所有的方法。

**2、抽象类要被子类继承，接口要被子类实现**。

**3、接口里面只能对方法进行声明，不能有实现；抽象类既可以对方法进行声明也可以对方法进行实现。**

4、抽象类里面的抽象方法必须全部被子类实现，如果子类不能全部实现，那么子类必须也是抽象类。

接口里面的方法也必须全部被子类实现，如果子类不能实现那么子类必须是抽象类。

（5、接口里面的方法只能声明，不能有具体的实现。这说明接口是设计的结果，抽象类是重构的结果。）

5、抽象类里面可以没有抽象方法。

6、如果一个类里面有抽象方法，那么这个类一定是抽象类。

7、抽象类中的方法都要被实现，所以抽象方法不能是静态的static，也不能是私有的private。

8、**接口（类）可以继承接口，甚至可以继承多个接口。但是类只能继承一个类。**

9、抽象级别（从高到低）：接口>抽象类>实现类。

10、抽象类主要是用来抽象类别，接口主要是用来抽象方法功能。当你关注事物的本质的时候，请用抽象类；当你关注一种操作的时候，用接口。

11、抽象类的功能应该要远多于接口，但是定义抽象类的代价较高。因为高级语言一个类只能继承一个父类，即你在设计这个类的时候必须要抽象出所有这个类的子类所具有的共同属性和方法；但是类（接口）却可以继承多个接口，因此每个接口你只需要将特定的动作方法抽象到这个接口即可。也就是说，接口的设计具有更大的可扩展性，而抽象类的设计必须十分谨慎。

1. Java中重载与重写的区别
2. 重载(Overloading)  
    （1）重载的时候，方法名要一样，但是参数类型和个数不一样，返回值类型可以相同也可以不相同。无法以返回型别作为重载函数的区分标准。  
    （2）[Java](http://java.chinaitlab.com/)的方法重载，就是在类中可以创建多个方法，它们具有相同的名字，但具有不同的参数和不同的定义。调用方法时通过传递给它们的不同参数个数和参数类型来决定具体使用哪个方法, 这就是多态性。

2、重写（Overriding）  
  （1）如果在子类中定义某方法与其父类有相同的名称、参数和返回类型，我们说该方法被重写 (Overriding)，则新方法将覆盖父类中原有的方法（）如需父类中原有的方法，可使用super关键字，该关键字引用了当前类的父类。）。在[Java](http://java.chinaitlab.com/)中，子类可继承父类中的方法，而不需要重新编写相同的方法，但有时子类并不想原封不动地继承父类的方法，而是想作一定的修改，这就需要采用方法的重写。

（2）、这体现了父类与子类之间的多态性，对父类的函数进行重新定义。方法重写又称方法覆盖。  
（3）子类函数的访问修饰权限不能少于父类的；动态绑定。