Java中接口学习

1. 接口本身就是一个抽象类；接口不可以实例化，
2. 接口中的方法都是抽象方法；抽象方法就是用来被继承的；所以一个类实现一个接口，必须重写接口中的所有方法，只有这样才可以实例化，否则这个子类也应该是抽象类。
3. **接口中的方法都是public 权限；注意一般情况下，不显式地声明权限都是default的，但是在接口中都是public类型的。**
4. 接口利用interface{ } 定义；
5. 接口中的成员修饰符都是固定的：（1）成员常量：public static final；

（2）成员函数 ：public abstract （3）**接口中的成员都是公共权限public的。**

简单说明一下：这些修饰符在定义接口中都可以省略，但是导致可读性比较差。

1. 虽然接口不可以实例化，即不可以生成对象，但是可以生成一个接口引用；接口引用可以指向任意实现此接口的所有类，相当于父类的引用可以指向子类。
2. 注意：抽象方法没有主体，不能带大括号；所以接口中的方法都是不能带大括号的，即使是空的也不可以。
3. 接口的实现利用implements 关键字。
4. 一个类只可以继承一个类，即一个类只有一个父类，Java只支持单继承，因为多继承会出现继承的不确定性；但是一个类可以实现多个接口；java中不支持多继承但是支持**多实现，多实现就是对C++的多继承的改良**；
5. 类与类之间是继承关系，并且是单继承的；接口与接口之间是继承关系，是多继承的关系，类与接口之间是实现关系，一个类可以实现多个接口。一个类在继承另外一个类的同时，可以同时实现多个接口。因此接口的出行避免了单继承的局限性。
6. 类与类之间的单继承，接口与接口的多继承，主要差别在于**有没有方法体**；接口中没有方法体，鉴于java对继承的改良，没有方法体的可以实现多继承。如一个类只可以继承另外一个类，因为另外一个类中有方法体，同时可以实现多个接口，因为这些接口中的方法都没有方法体（抽象方法）。**Java之所以对C++的多继承进行改良，是因为多继承导致继承了很多重名的方法，这些重名方法各自有各自的方法体，从而导致调用的不确定性。**
7. **面试的时候，若问java支持多继承吗？**

**答：java对C++的多继承进行了改良，只不过是java中的多继承是利用多实现的方法来体现，一个类可以继承另外一个类的同时可以实现多个接口，而接口之间又支持多继承，java中多继承的关键在于方法中方法体是否存在。**

1. 一个接口可以继承多个接口，接口之间的继承仍然利用关键字extends 实现。
2. 当一个类中的方法都是抽象的时候，可以将该抽象类用另外一种形式定义和表示，就是接口interface。**接口中的方法都是抽象方法**。
3. **广义接口的特点：**
4. **接口是对外暴露的规则；**
5. **接口是程序的功能扩展；**
6. **接口的出现降低了耦合性；**
7. **接口可以用来多实现；**
8. **类与接口是实现接口，而且类继承另外一个类的同时可以实现多个接口；**
9. **接口与接口之间又有继承的关系。**
10. **Java中接口就是定义规则，而实现接口的类就是在实现规则。**
11. **接口与抽象类的区别：**

**答： 相同点：都是不断向上抽取而来的；**

**不同点：（1）抽象类需要被继承，而且只能被单继承；**

**接口需要被实现，而且可以被类多实现；同时接口之间可以多继承；**

**（2）抽象类中既可以定义抽象方法也可以定义非抽象方法，子类继承抽象类，必须实现其中的抽象方法，只有这样才可以实例化，而且可以直接使用抽象类中的非抽象方法；接口中只可以定义抽象方法，必须由类去实现。**

**（3）抽象类的继承是在定义该体系的基本共性内容；**

**接口的实现是在定义体系的额外功能。**

**显然如果是一类事物的共性内容就定义一个抽象类，如果是这类事物的额外功能，定义为接口，这个额外功能可能由很多类事物都共有的，哪个有哪个去实现这个接口即可。**

1. **接口（interface）再次学习：**

**接口就是定义了调用对象的标准，只是标准，具体实现类如何实现，接口并没有做出规定。**

1. **接口是一种抽象类，接口中的方法都是抽象方法（无需特别注明abstract），接口中的方法只有方法定义没有方法体；**
2. **接口中的方法的权限都是public的，意义何在？**
3. **类和接口的关系：可以把接口看做是特殊的抽象类，一般的抽象类中既可以包括具体方法也可以包括抽象方法；但是接口中只可以包括抽象方法。类可以实现接口。实现时利用关键字implements进行，代表某个类对接口的实现。一个类要实现一个接口，该类要实现该接口中所声明的所有方法。如果该类是抽象类，那就无需实现该接口中的方法。一个类可以同时继承一个类，也实现多个接口，多个接口之间用逗号隔开，例如： class BB extends AA implements MySurface1,MySurface2 。**
4. **抽象类不可以生成对象，同样接口也不可以生成对象；但是可以利用具体类去实现接口。（类实现（implements）接口，实现是一种特殊的继承，所以接口看做是一种“父类”，实现类是子类。）**
5. **由于接口中的方法都是抽象方法，所以实现接口的类中必须重写接口中的所有方法。**
6. **一个类只能继承一个类，但是一个类可以实现多个接口（也是多态性的表现）。如手机既可以支持WiFi上网，又支持4G，3G，2G等上网。既然一个类实现了多个接口，那么此实现类向上类型转换时就有了多种选择。**
7. **一个接口可以继承多个接口。类实现继承接口时，必须实现所有的接口中的方法。**
8. **接口的使用：**
9. **工厂方法模式：**
10. **接口（interface）：接口的地位和类一样，接口中的方法都是抽象方法。利用interface关键字进行声明。在声明接口中的方法时，既可以利用abstract进行声明也可以不声明，因为已经规定了接口中的方法一定是抽象的，一般情况下需要省略abstract关键字。**
11. **接口不可以实例化，只是利用类进行实现。**
12. **类和接口的关系：可以把接口看做是特殊的抽象类，一般的抽象类中既可以包括具体方法也可以包括抽象方法；但是接口中只可以包括抽象方法。类可以实现接口。实现时利用关键字implements进行，代表某个类对接口的实现。一个类要实现一个接口，该类要实现该接口中所声明的所有方法。如果该类是抽象类，那就无需实现该接口中的方法。一个类可以同时继承一个类，也实现多个接口，多个接口之间用逗号隔开，**
13. **例如： class BB extends AA implements MySurface1,MySurface2 。**
14. **建一个类去实现一个接口，必须实现该接口的所有方法(抽象方法)**