1. 使用說明：

　 1.接口使用interface來定義

\* 2.Java中，接口和類是並列的兩個結構

\* 3.如何定義接口：定義接口中的成員

\* 3.1 JDK7及以前：只能定義全局常量和抽象方法

\* >全局常量：都是public static final的,但是書寫時，可以省略不寫

\* >抽象方法：都是public abstract的,書寫時，可以省略不寫

\*

\* 3.2 JDK8：除了定義全局常量和抽象方法之外，還可以定義靜態方法、默認方法

\* 4. 接口中不能定義構造器的！意謂著接口不可以實例化

\* 5. Java開發中，接口通過讓類去實現(implements)的方式來使用.

\* 如果實現類實現(重寫)了接口中的所有抽象方法，則此實現類就可以實例化

\* 如果實現類沒有實現(重寫)接口中所有的抽象方法，則此實現類仍為一個抽象類

\* 6. Java類可以實現多個接口 --->彌補了Java單繼承性的局限性

\* 格式：class AA extends BB implements CC,DD,EE

\* 7. 接口與接口之間可以繼承，而且可以多繼承

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 8. 接口的具體使用，體現多態性

\* 9. 接口，實際上可以看做是一種規範

2. 舉例:



**class** Computer{

**public** **void** transferData(USB usb){//USB usb = new Flash(); //多態性

usb.start();

System.***out***.println("具體傳輸數據的細節");

usb.stop();

}

}

**interface** USB{

//常量：定義了長、寬、最大最小的傳輸速度等

**void** start();

**void** stop();

}

**class** Flash **implements** USB{

@Override

**public** **void** start() {

System.***out***.println("U盤開啟工作");

}

@Override

**public** **void** stop() {

System.***out***.println("U盤結束工作");

}

}

**class** Printer **implements** USB{

@Override

**public** **void** start() {

System.***out***.println("打印機開啟工作");

}

@Override

**public** **void** stop() {

System.***out***.println("打印機結束工作");

}

}

體會:

\* 1.接口使用上也滿足多態性

\* 2.接口，實際上就是定義了一種規範

\* 3.開發 中，體會面向接口編程！

3. 體會面向接口編程的思想



面向接口編程:我們在應用程序中,調用的結構都是JDBC中定義的接口,不會出現具體某一個數據庫廠商的API。

4. Java8中關於接口的新規範:

知識點1：接口中定義的靜態方法，只能通過接口來調用。(接口名.方法名)

知識點2：通過實現類的對象，可以調用接口中的默認方法。

如果實現類重寫了接口中的默認方法，調用時，仍然調用的是重寫以後的方法

知識點3：如果子類(或實現類)繼承的父類和實現的接口中聲明了同名同參數的默認方法，

那麼子類在沒有重寫此方法的情況下，默認調用的是父類中的同名同參數的方法。-->類優先原則

知識點4：如果實現類實現了多個接口，而這多個接口中定義了同名同參數的默認方法，那麼在實現  
 類沒有重寫此方法的情況下，報錯。-->接口衝突。

這就需要我們必須在實現類中重寫此方法

知識點5：如何在子類(或實現類)的方法中調用父類、接口中被重寫的方法

**public** **void** myMethod(){

method3();//調用自己定義的重寫的方法

**super**.method3();//調用的是父類中聲明的

CompareA.*method1*();//調用接口中定義的靜態方法

//調用接口中的默認方法

CompareA.**super**.method3();

CompareB.**super**.method3();

}

5.面試題

抽象類和接口的異同?

相同點:不能實例化、都可以包含抽象方法

不同點:

1. 把抽象類和接口(JDK7.8.9)的定義、內部結構解釋說明
2. 類:單繼承性 接口:多繼承  
   類與接口:多實現