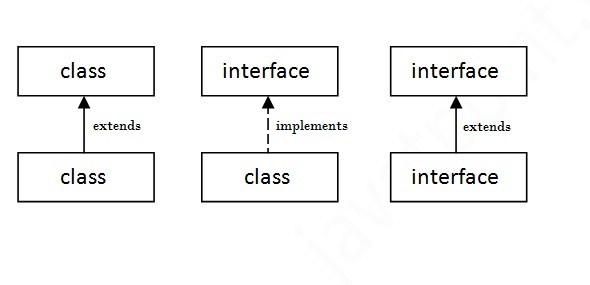
Nama : sanjaya m faishal  
kelas : Tif 15 b   
npm : 15111178  
  
 interface

Interface adalah sebuah tipe referensi pada Java. Interface secara struktur serupa dengan class. Isi dari interface adalah method abstract, artinya method hanya dideklarasikan tidak ditulis secara utuh. Deklarasi method pada interface serupa dengan method pada [**class abstract**](http://onestringlab.com/2015/12/10/penggunaan-perintah-abstract-final-dan-static-pada-java/).

Variable pada interface akan memiliki aturan static dan final sedangkan method akan memiliki aturan public dan abstract. Hal ini terjadi setelah proses kompilasi seperti yang terlihat pada gambar.

## C:\Users\sanjaya m faishal\Downloads\interface.jpgHubungan Antara Class dan Interface

Seperti yang terlihat pada gambar dibawah ini, sebuah class melakukan extends terhadap class  lain, sebuah interface melakukan extends terhadap interface lain, namun sebuah **class melakukan implements pada interface**.  
  
  
***Kenapa Kita Memakai Interface?***  
  
Kita akan menggunakan interface jika kita ingin class yang tidak berhubungan mengimplementasikan method yang sama. Melalui interface-interface, kita dapat menangkap kemiripan diantara class yang tidak berhubungan tanpa membuatnya seolah - olah class yang berhubungan.  
  
Mari kita ambil contoh class Line dimana berisi method yang menghitung panjang dari garis dan membandingkan object Line ke object dari class yang sama. Sekarang, misalkan kita punya class yang lain yaitu MyInteger dimana berisi method yang membandingkan object MyInteger ke object dari class yang sama.

**Membuat Interface**  
  
Untuk membuat interface,   
  
--------------------------------------------------------  
public interface [InterfaceName]  
{  
//beberapa method tanpa isi  
}  
  
--------------------------------------------------------  
  
Sebagai contoh, mari kita membuat sebuah interface yang mendefinisikan hubungan antara  
dua object menurut urutan asli dari object.  
  
--------------------------------------------------------  
public interface Relation  
  
  
{  
  
  
public boolean isGreater( Object a, Object b);  
  
  
public boolean isLess( Object a, Object b);  
  
  
public boolean isEqual( Object a, Object b);  
  
  
}  
  
--------------------------------------------------------  
  
Sekarang, penggunaan interface, kita gunakan kata kunci implements. Contohnya,  
  
---------------------------------------------------------  
/\*\*  
  
  
\* Class ini mendefinisikan segmen garis  
\*/  
public class Line implements Relation  
{  
  
  
private double x1;  
private double x2;  
private double y1;  
private double y2;  
  
  
public Line(double x1, double x2, double y1, double y2){  
this.x1 = x1;  
this.x2 = x2;  
this.y1 = y1;  
this.y2 = y2;  
  
  
}  
public double getLength(){  
double length = Math.sqrt((x2-x1)\*(x2-x1) +  
(y2-y1)\* (y2-y1));  
return length;  
}  
  
  
public boolean isGreater( Object a, Object b){  
double aLen = ((Line)a).getLength();  
double bLen = ((Line)b).getLength();  
return (aLen > bLen);  
  
  
}

contoh syntax interface yang lain  
  
  
public interface TampilLaporan {

    public void tampilWeb();

    public void tampilMobile();

}

public class LaporanTahunan implements CetakLaporan, TampilLaporan{

    public void cetakA4(){

        System.out.println("Cetak Laporan di A4");

    }

    public void cetakA3(){

        System.out.println("Cetak Laporan di A3");

    }

    public void tampilWeb(){

        System.out.println("Tampil di Monitor");

    }

    public void tampilMobile(){

        System.out.println("Tampil di Handphone");

    }

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        LaporanTahunan lt = new LaporanTahunan();

        lt.cetakA3();

        lt.cetakA4();

        lt.tampilMobile();

        lt.tampilWeb();

    }

}

}

Contoh kode program interface melakukan multiple pewarisan.

public interface Laporan extends CetakLaporan, TampilLaporan{

    public void exportExcel();

}