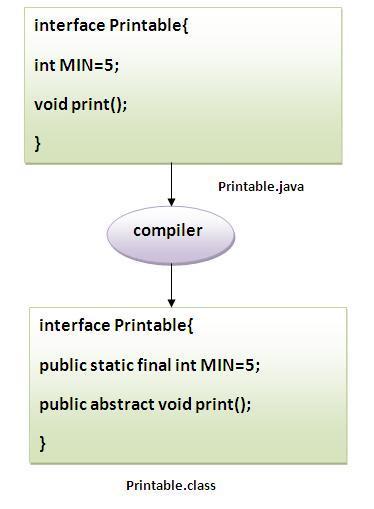
Penjelasan interface dan contoh penggunaan

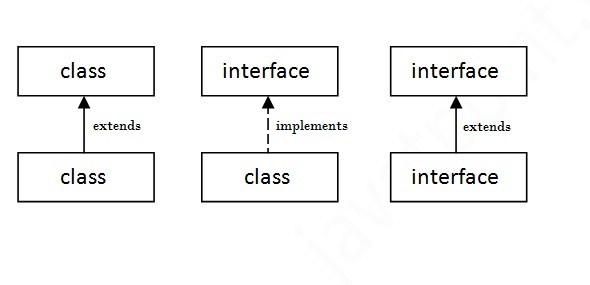
Interface adalah sebuah tipe referensi pada Java. Interface secara struktur serupa dengan class. Isi dari interface adalah method abstract, artinya method hanya dideklarasikan tidak ditulis secara utuh. Deklarasi method pada interface serupa dengan method pada [**class abstract**](http://onestringlab.com/2015/12/10/penggunaan-perintah-abstract-final-dan-static-p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

Perubahan Aturan Setelah Kompilasi Pada Interface

**Hubungan 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**.

Hubungan Interface dan Class

Perhatikan kode interface CetakLaporan:

public interface CetakLaporan {

    public void cetakA4();

    public void cetakA3();

}

Interface CetakLaporan diimplementasikan pada class LaporanTahunan

public class LaporanTahunan implements CetakLaporan{

    public void cetakA4(){

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

    }

    public void cetakA3(){

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

    }

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        LaporanTahunan lt = new LaporanTahunan();

        lt.cetakA3();

        lt.cetakA4();

    }

}

Interface adalah sebuah tipe referensi pada java. Secara Struktur hampir sama dengan class. Ada beberapa aturan dalam penulisan interface :

1. Modifier method hanya boleh public, abstract atau default
2. Pada class yang mengimplement, modifier method – method hanya boleh public
3. Jumlah parameter method interface harus sama dengan class yang mengimplemetnya (Pas kita override pasti langsung sama parameternya, jangan diganti!)
4. Tidak boleh ada method concrete di interface. Maksudnya begini, jika kita menambahkan method seperti ini public void output(){System.out.println();}, maka akan error.
5. Tidak boleh ada constructor
6. Interface tidak bisa di instanisasi, TAPI bisa di instanisasi melalui class yang meng implementnya