

Jawaban

1. Inheritance adalah konsep dalam OOP yang memungkinkan sebuah class untuk 'mewarisi' karakteristik atau atribut dari kelas lain yang biasa disebut dengan parent class. Dalam pemrograman java, penggunaan konsep inheritance adalah dengan keyword 'Extends' dan 'Super'.
2. kelas Abstract : class yang tidak memiliki objek atau objeknya tidak dapat dibuat secara langsung. Abstrak dapat memaksakan suatu atribut ataupun method kepada sebuah kelas lain
Kelas Interface : class tipe khusus yang hanya berisi deklarasi method tanpa implementasi. Interface biasanya digunakan untuk menentukan "kontrak" yang harus dimiliki oleh kelas yang mengimplementasikan interface tersebut.
3. aturan terkait pembuatan abstract adalah
 - Deklarasi kelas abstrak perlu menggunakan keyword Abstract di depan kelasnya
 - Kelas abstrak dapat memiliki metode yang tidak memiliki isi dan hanya dideklarasikan dan metode dengan implementasi.aturan terkait pembuatan interface adalah
 - Deklarasi kelas Interface perlu menggunakan keyword interface di depan kelasnya
 - Semua metode dalam interface secara default bersifat **public** dan *abstract* (tanpa implementasi).
 - Kelas dapat mengimplementasikan lebih dari satu interface akan tetapi hanya dapat mewarisi satu kelas.
4. Tujuan dari konsep kelas abstract adalah agar suatu objek memiliki perilaku atau atribut mendasar yang sama dan untuk mengurangi redundansi dalam pendefinisian dari sebuah method, dimana method dapat didefinisikan dalam suatu class abstrak untuk dipakai dalam semua object yang dibentuk dari kelas tersebut.

Tujuan dari konsep kelas Interface adalah untuk mendefinisikan kemampuan yang bisa dimiliki oleh banyak kelas berbeda, meskipun tidak terkait secara langsung melalui pewarisan.

5. contoh coding sederhana implementasi dari abstract

```
import java.util.ArrayList;

import javax.print.DocFlavor.STRING;

//abstract

public class kelasTerbuka { //public class jd bsa dipake di package manapun as long di import
```

```

//parent class
public abstract class Hero{
    public String name;

    public Hero(String name){
        this.name = name;
    }

    void display(){
        System.out.println("nama Hero : "+this.name);
    }
}

//child class
class HeroStrength extends Hero{
    String type ="power";

    HeroStrength(String name){//klo di parent ada
//constructor di sini jg hrs
        super(name); //manggil constructor di parent
    }
}

public static void main(String[] args) {
    kelasTerbuka outer = new kelasTerbuka();
    HeroStrength h2 = outer.new HeroStrength("mr
incredible");
    h2.display();
}
}

```

dan interface!

```

import java.util.ArrayList;

import javax.print.DocFlavor.STRING;

//interface

```

```
public class kelasTerbuka { //public class jd bsa dipake di package manapun  
as long di import  
    interface Terbang {  
        void terbang(); // Abstract method  
    }  
  
    interface Berenang {  
        void berenang(); // Abstract method  
    }  
  
    class Bebek implements Terbang, Berenang {  
        public void terbang() {  
            System.out.println("Bebek terbang");  
        }  
  
        public void berenang() {  
            System.out.println("Bebek berenang");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    kelasTerbuka outer = new kelasTerbuka();  
  
    Bebek b1 = outer.new Bebek();  
    b1.berenang();  
}  
}
```