

Latihan :

1. Class TrafficLight berikut adalah simulasi dari lampu lalu lintas. Coba lakukan setting pada TrafficLight tsb dengan menyalakan lampu merah sebanyak 5 kali, selanjutnya lampu kuning dinyalakan 2 kali, diakhiri dengan menyalakan lampu hijau 10 kali.

Petunjuk : Pada main method lakukan instansiasi obyek yang mengacu ke class TrafficLight. Untuk akses ketiga lampu dapat dilakukan melalui method turnLight().

```
public class TrafficLight{
    boolean bred,bgreen,byellow;
    void turnLight(int color){
        bred=bgreen=byellow=false;
        switch(color){
            case 0: //merah
                bred = true;
                break;
            case 1: //hijau
                bgreen = true;
                break;
            case 2: //kuning
                byellow = true;
                break;
        }
    }
    public static void main(String args[])
    {
        // instansiasi obyek
        Dan tampilkan
    }
}
```

2. Tuliskan output dari kode program berikut :

```
class Point{
    int x=1,y=1;
    void getXY(int x, int y){
        x = this.x - x;
        y = this.y + y;
        System.out.println(x);
        System.out.println(y);
    }

    public static void main(String args[]){
        Point titik = new Point();
        titik.getXY(2,3);
    }
}
```

3. overload constructor

perhatikan kode program berikut :

```
class Employee{
    String nama;
    Employee(){
        System.out.println("Default constructor");
    }
    Employee(String n){
        this.nama = n;
        System.out.println("constructor dengan 1 parameter");
    }
    void getDetails(){
        System.out.println(nama);
    }
    public static void main(String args[])
    {
        //Add some code here
        ....
    }
}
```

- a. lakukan pemanggilan constructor yang memiliki satu parameter melalui obyeknya.
- b. lakukan overload constructor, tambahkan constructor dengan 3 parameter, yaitu : nama, jabatan, dan nomor induk pegawai. Kemudian lakukan pemanggilan constructor tersebut melalui obyeknya.

4. Tuliskan output yang dihasilkan dari runtime kode program berikut ini :

```
public class Employee{
    Employee()
    {
        System.out.println("Konstruktor Employee() dijalankan");
    }
}
class Manager extends Employee{
    Manager()
    {
        System.out.println("Konstruktor Manager() dijalankan");
    }
    public static void main(String args[]){
        Manager e = new Manager();
    }
}
```

5. Contoh Override method

```
public class Point{
    int x=2,y=4;
    int deltaX = 1, deltaY = 1;
    int getX()
    {
        return x+deltaX;
    }
    int getY()
    {
        return y+deltaY;
    }
}
class Shape extends Point{
    int x=5, y=7;
    int testSuper()
    {
        System.out.println("(x,y) = " + x + ", " + y);
        System.out.println("(super.x,super.y) = " + super.x + ", " + super.y);
        System.out.println();
        System.out.println("getX() = " + getX());
        System.out.println("super.getX() = " + super.getX());
        return 0;
    }
    int getX()
    {
        return this.x;
    }
    int getY()
    {
        return this.y;
    }
    public static void main(String args[])
    {
        Shape shp = new Shape().testSuper();
    }
}
```

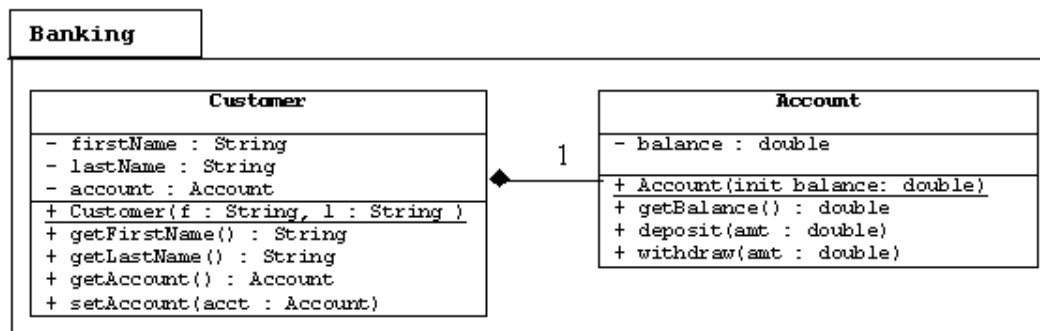
6. Method accessor & mutator

Dari kode program berikut tentukan yang mana termasuk method accessor dan mutator, sebutkan dan berikan alasannya.

```
public class Karyawan{
    // instance variabel
    private String nama;
    private double gaji;
    void setNama(String nm){
        nama = nm;
    }
    void setGaji(double gj){
        gaji = gj;
    }
    String getNama(){
        return nama;
    }
    double getGaji(){
        return gaji;
    }
}
```

Tugas Praktikum :

1. Buatlah program dengan membaca diagram class berikut ini :



Keterangan :

Class Account merupakan superclass dari class Customer

Variabel :

firstname : menyimpan nama depan customer

lastname : menyimpan nama belakang customer

Method :

getBalance : untuk mendapatkan nilai saldo saat ini.

deposit : untuk menambah jumlah saldo sejumlah value yang dikirim pada parameter. Jika saldo berjumlah Rp. 500.000,- atau kelipatannya maka otomatis akan mendapatkan bunga sebesar 0.5% dari jumlah saldo.

withdraw : untuk mengurangi jumlah saldo sejumlah value yang dikirim pada parameter. Untuk penarikan diatas Rp. 2.000.000,- akan dikenai biaya Rp. 15.000,-

getFirstname : untuk mendapatkan nama depan dari customer

getLastname : untuk mendapatkan nama belakang dari customer

getAccount : method accesor untuk variabel account

setAccount : method mutator untuk variabel account

perintah :

a. buatlah main method pada class account. Buat akun untuk beberapa nasabah berikut :

Firstname	LastName	Saldo awal	Saldo awal+Bunga	Saldo terakhir
Joko	Kusumo	1.000.000	?	?
Indah	Septiarini	350.000	?	?
Sujarwo	Adiwijoyo	5.250.000	?	?
Muhammad	Hasan	50.000	?	?

b. Nasabah bernama Joko menabung sebesar 1.750.000, kemudian melakukan pengambilan sebesar 2.000.000

c. Nasabah bernama Indah menabung sebesar 150.000, dan mengambil uang sebesar 500.000

d. Nasabah bernama Sujarwo mengambil uang sebesar 6.000.000

e. Nasabah bernama Muhammad mengambil uang sebesar 500.000

petunjuk :

tambahkan perintah untuk mengecek saldo nasabah. Jika saldo tidak cukup, maka tampilkan pesan peringatan.

2. Interface

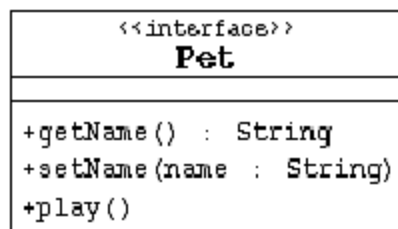
Buatlah sebuah interface dengan nama Pet yang memiliki 3 method yaitu :

getName() : memiliki returnType bertipe string

setName(String name) : memiliki 1 parameter bertipe string

Play() : tidak memiliki returnType

Lebih jelasnya perhatikan class diagram dibawah ini :



Perintah : implemetasikan interface tersebut ke Class MyPet. Tambahkan private variabel bernama "name" pada class tersebut. Hasil overriding dari method play() diisi dengan perintah cetak kelayar "<name> sedang berguling-guling, lucu!".