

Nama : Dhanar Agastya Rakalangi
NRP : 5002221075

1. (a) Jelaskan yang dimaksud dengan inheritance, encapsulation dan polymorphism dalam konsep Pemrograman Berorientasi Objek

Jawab :

- Inheritance : Konsep di mana sebuah class dapat mewarisi properti dan metode dari class lain. Dengan menggunakan inheritance, sebuah class dapat mengambil karakteristik dari class induknya tanpa perlu menulis ulang kode yang sama
- Encapsulation : Konsep di mana data (variabel) dan metode yang beroperasi pada data tersebut dikemas dalam sebuah class. Dengan menggunakan encapsulation, data di dalam class hanya dapat diakses dan diubah oleh metode yang ada di dalam class tersebut
- Polymorphism : Konsep di mana sebuah objek dapat memiliki berbagai bentuk atau tipe, dan dapat berperilaku berbeda tergantung konteksnya. Polymorphism memungkinkan objek dari class-class yang berbeda untuk diakses dan diproses dengan cara yang sama, sehingga memungkinkan pemrogram untuk menulis kode yang lebih fleksibel dan reusable.

- (b) Apa yang anda ketahui tentang garbage collection dalam Java, beri penjelasan

Jawab :

Garbage collection merupakan proses otomatis yang dilakukan oleh Java Virtual Machine (JVM) untuk mengelola memori yang digunakan dalam program Java. Garbage collection bertugas untuk mencari dan menghapus objek-objek yang tidak lagi digunakan atau direferensikan oleh program agar memori tersebut dapat digunakan kembali.

Proses garbage collection dilakukan dengan mengidentifikasi objek-objek yang sudah tidak terpakai dan melepaskan memori yang dialokasikan untuk objek-objek tersebut. JVM akan secara otomatis melakukan garbage collection secara terjadwal atau ketika sisa memori sudah hampir penuh.

- (c) Jelaskan perbedaan method overloading dan method overriding.

Jawab :

- Method overloading adalah kemampuan untuk mendefinisikan beberapa method dengan nama yang sama di sebuah class, namun memiliki parameter yang berbeda.
- Sementara itu, Method overriding adalah ketika subclass mendefinisikan kembali (override) method yang sudah ada di superclass dengan mengganti implementasi method tersebut.

2. Lengkapilah potongan program berikut :

```
class Account {  
  
    private long acc_no;  
    private String name,email;  
    private int amount;  
  
    public long getAcc_no() {  
        .....  
    }  
  
    public void setAcc_no(long acc_no) {  
        .....  
    }  
    public String getName() {  
        .....  
    }  
    public void setName(String name) {  
        .....  
    }  
    public String getEmail() {  
        .....  
    }  
}
```

Lanjutan

```
    public void setEmail(String email) {  
        .....  
    }  
    public int getAmount() {  
        .....  
    }  
    public void setAmount(int amount) {  
        .....  
    }  
}  
  
public class UjiEncapsulation {  
    public static void main(String[] args) {  
        .....  
    }  
}
```

Agar keluaran dari program adalah sebagai berikut :

123456 Maudy Ayunda mayunda@gmail.com 50000000

Jawab :

```
class Account {  
    private long acc_no;  
    private String name, email;  
    private int amount;  
  
    public long getAcc_no() {  
        return acc_no;  
    }  
  
    public void setAcc_no(long acc_no) {  
        this.acc_no = acc_no;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

```

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public int getAmount() {
        return amount;
    }

    public void setAmount(int amount) {
        this.amount = amount;
    }
}

public class UjiEncapsulation {
    public static void main(String[] args) {
        Account acc = new Account();
        acc.setAcc_no(123456);
        acc.setName("Maudy Ayunda");
        acc.setEmail("mayunda@gmail.com");
        acc.setAmount(50000000);

        System.out.println(acc.getAcc_no() + " " + acc.getName() + " " +
            acc.getEmail() + " " + acc.getAmount());
    }
}

```

3. Apa yang dikerjakan oleh program berikut (jelaskan tahap-demi-tahap) sehingga menghasilkan keluaran yang benar. (**HINT** : perhatikan tipe argument dari method **cetak**. Class Object merupakan class tertinggi)

```
package ets_2019;
public class ETS_2019{
    public static void main(String[] args) {
        Gagak BG = new Gagak();
        cetak (BG);
        cetak(new Burung());
        cetak(new Hewan());
    }
    public static void cetak(Hewan x) {
        System.out.println(x.getket() + "---" + x.warna);
    } }
class Gagak extends Burung{
    public Gagak() {
        System.out.println("Objek Gagak diciptakan");
        warna = "COKLAT";
    } }
}
```

```
class Burung extends Hewan{
    public String getket() {
        return "Burung";
    } }
class Hewan {
    protected String warna;

    public Hewan() {
        System.out.println("Objek Hewan diciptakan");
        warna = "HITAM";
    }
    public String getket() {
        return "HEWAN";
    }
}
```

Jawab :

- Di dalam method main, pertama-tama membuat objek BG dari class Gagak.
- Ketika objek BG diciptakan, constructor dari class Gagak dipanggil dan mencetak "Objek Gagak diciptakan". Selain itu, nilai warna dari objek BG diinisialisasi menjadi "COKLAT".
- Kemudian memanggil method cetak dengan parameter objek BG. Dalam method cetak, akan mencetak hasil dari get.ket() dari class Gagak ("Burung") dan nilai dari warna dari objek BG ("COKLAT").

- Selanjutnya, membuat objek baru dari class Burung tanpa nama dan langsung memanggil method cetak dengan objek tersebut sebagai parameter. Dalam method cetak, akan mencetak hasil dari get.ket() dari class Burung ("Burung") dan nilai warna dari objek tersebut (yang diinisialisasi dari constructor class Hewan dengan warna HITAM).
- Terakhir, membuat objek baru dari class Hewan tanpa nama dan langsung memanggil method cetak dengan objek tersebut sebagai parameter. Dalam method cetak, akan mencetak hasil dari getket() dari class Hewan ("HEWAN") dan nilai warna dari objek tersebut (yang diinisialisasi dari constructor class Hewan dengan warna HITAM).

Output :

Objek Gagak diciptakan

Burung — COKLAT

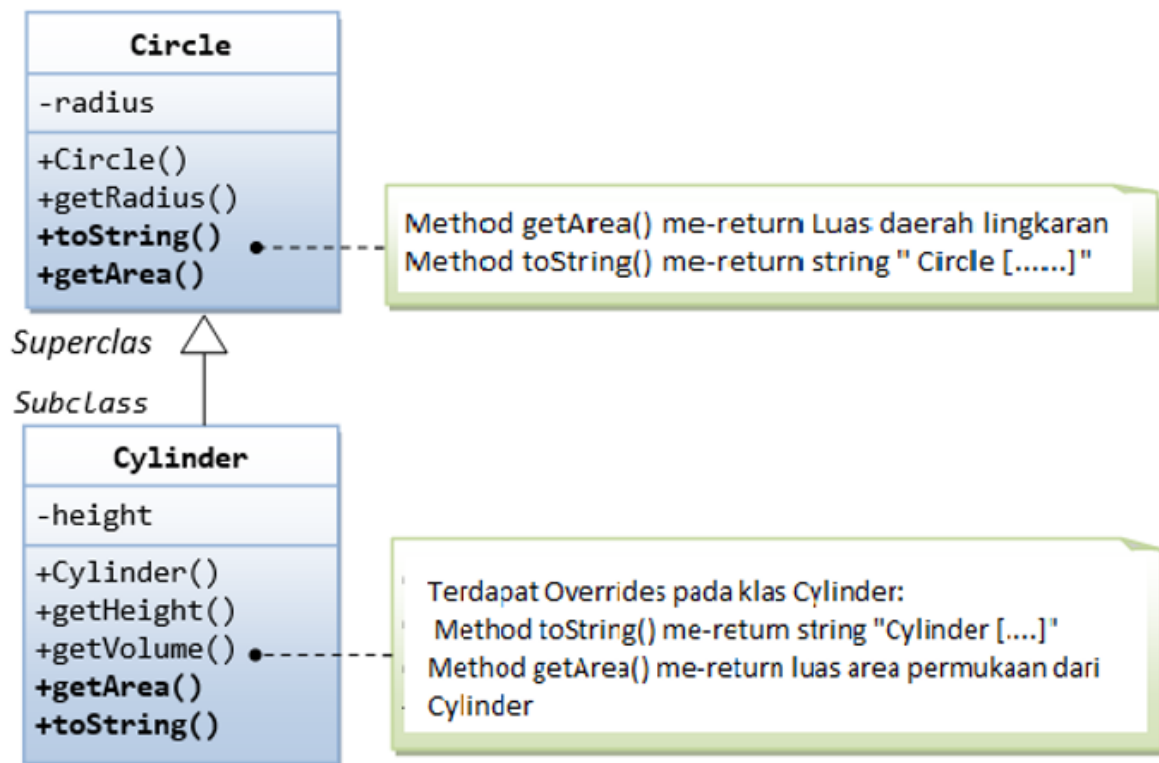
Objek Hewan diciptakan

Burung — HITAM

Objek Hewan diciptakan

HEWAN — HITAM

4. Perhatikan class Diagram dalam UML berikut dengan sangat seksama. Implementasikan class diagram tersebut. Buat contoh main program dengan memanfaatkan method-method yang ada.



Jawab :

```
package Latihan.EAS.PBO;
```

```
import java.util.*;
```

```
public class No4 {
```

```
    public static void main(String [] args) {
```

```
        Scanner input = new Scanner(System.in);
```

```
        System.out.print("Masukkan radius: ");
```

```
        int r = input.nextInt();
```

```
        System.out.print("Masukkan height: ");
```

```
        int h = input.nextInt();
```

```
        Cylinder obj = new Cylinder(r,h);
```

```
        System.out.println(obj.toString());
```

```
        System.out.println("Luas Permukaan: " + obj.getArea());
```

```
        System.out.println("Volume: " + obj.getVolume());
```

```
        Circle obj2 = new Circle(r);
```

```

        System.out.println(obj2.toString());
        System.out.println("Luas:-" + obj2.getArea());
    }
}

class Circle {
    private double radius;

    public Circle(int x){
        radius = x;
    }
    public double getRadius(){
        return radius;
    }
    public String toString(){
        return "Circle [] -dengan-radius-" + radius;
    }
    public double getArea(){
        return Math.PI*radius*radius;
    }
}

class Cylinder extends Circle{
    private double height;
    public Cylinder(int x, int t){
        super(x); height = t;
    }
    public double getHeight(){
        return height;
    }
    public double getVolume (){
        return Math.PI*getRadius()*getRadius()*height;
    }
    public String toString (){
        return "Cylinder [] -dengan-radius-" + getRadius() + "-dan"
            + "-tinggi-" + height;
    }
}

```



```

    public double getArea() {
        return 2*Math.PI*getRadius()*(getRadius()+height);
    }
}

```

5. Bangunlah sebuah class bernama bilPecahan dengan spesifikasi sebagai berikut

bilPecahan
<p>- pembilang : int - penyebut : int</p> <hr/> <p>+bilPecahan(); +bilPecahan(int a, int b); +bilPecahan jumlah(bilPecahan a); +bilPecahan kali(bilPecahan a); +String bentukCampuran(); +void infoPecahan();</p>

Keterangan :

Metode bentuk campuran akan mengembalikan nilai pecahan dalam bentuk campuran bertipe string, misal $\frac{5}{3}$ akan diubah menjadi $1\frac{2}{3}$, apabila pembilang lebih besar daripada penyebut.

Jawab :

```
package Latihan_EAS_PBO;
```

```

class bilPecahan {
    private int pembilang;
    private int penyebut;

    // konstruktor
    public bilPecahan(int pembilang, int penyebut) {
        this.pembilang = pembilang;
        this.penyebut = penyebut;
    }

    // getter
    public int getPembilang() {
        return pembilang;
    }

    public int getPenyebut() {
        return penyebut;
    }
}

```

```
}
```

```
// penjumlahan
```

```
public bilPecahan jumlah(bilPecahan a) {  
    int newPembilang = this.pembilang * a.getPenyebut()  
        + this.penyebut * a.getPembilang();  
    int newPenyebut = this.penyebut * a.getPenyebut();  
    return new bilPecahan(newPembilang, newPenyebut);  
}
```

```
// perkalian
```

```
public bilPecahan kali(bilPecahan a) {  
    int newPembilang = this.pembilang * a.getPembilang()  
        ;  
    int newPenyebut = this.penyebut * a.getPenyebut();  
    return new bilPecahan(newPembilang, newPenyebut);  
}
```

```
// bentuk campuran
```

```
public String bentukCampuran() {  
    if (pembilang >= penyebut) {  
        int integerPart = pembilang / penyebut;  
        int remainder = pembilang % penyebut;  
        return integerPart + "-" + remainder + "/" +  
            penyebut;  
    } else {  
        return pembilang + "/" + penyebut;  
    }  
}
```

```
// setter
```

```
public void setPembilang(int pembilang) {  
    this.pembilang = pembilang;  
}
```

```
public void setPenyebut(int penyebut) {
```

```
        this.penyebut = penyebut;
    }

    // info pecahan
    public void infoPecahan() {
        System.out.println("Pembilang: -" + pembilang);
        System.out.println("Penyebut: -" + penyebut);
    }
}
```

```
public class No5 {  
    public static void main(String args[]) {  
        bilPecahan p1 = new bilPecahan(3, 2);  
        bilPecahan p2 = new bilPecahan(1, 4);  
        bilPecahan hasilJumlah = p1.jumlah(p2);  
        System.out.println("Hasil-Penjumlahan: -" +  
            hasilJumlah.bentukCampuran());  
        bilPecahan hasilKali = p1.kali(p2);  
        System.out.println("Hasil-Perkalian: -" + hasilKali.  
            bentukCampuran());  
    }  
}
```