MODUL 14

STATIC MODIFIER, INHERITANCE, AND POLIMORPHISM

Adam Arthur Faizal

M3119001

TI A

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS SEBELAS MARET

SURAKARTA

2020

MODUL 14. STATIC MODIFIER, INHERITANCE, POLIMORPHISM

Capaian Pembelajaran Praktikum:

- Menggunakan static modifier
- Menerapkan konsep inheritance dalam program
- Menerapkan konsep polimorfisme dalam program

Tools:

- Java Development Kit (JDK)
- Eclipse

Terminologi:

Isikan terminology yang sesuai untuk definisi dibawah ini:

[Static Modifier] Is a keyword that makes a variable, method, or inner class available without first creating an instance of a variable.

[Inheritance] The concept in object-oriented programming that allows classes to gain methods and data by extending another classes fields and methods.

[Encapsulation] A programming philosophy that promotes protecting data and hiding implementation in order to preserve the integrity of data and methods.

[Super] A keyword that allows subclasses to access methods, data, and constructors from their parent class.

[Protected] Visible to the package where it is declared and to subclasses in other packages.

[Subclass] Classes that are more specific subsets of other classes and that inherit methods and fields from more general classes.

[Overriding Method] Implementing methods in a subclass that have the same prototype (the same parameters, method name, and return type) as another method in the superclass.

TRY IT / SOLVE IT:

1. Buat sebuah class Turtle dan class DriverTurtle sbb: Amati kesalahan/error yang terjadi pada class DriverTurtle, jelaskan penyebab error dan coba anda perbaiki.

```
public class Turtle {
    public static String food = "Turtle Feed";
    private int age;
    private int tankNum;
    public static int numTanks = 3;
    public Turtle(int age){
        this.age = age;
        tankNum = (int)((Math.random()*numTanks)+1);
    }
    //public void swim(){}
    public int getAge(){
        return age;
    }
    public int getTankOfResidence(){
        return tankNum;
    }
    public static String fishTank() {
        return "I have " + numTanks + " fish tanks.";
    }

}
```

```
public class DriverTurtle {
    public static void main(String[] args) {
        Turtle T1=new Turtle(1);
        Turtle T2=new Turtle(2);
        System.out.println("Jumlah Tangki Total adalah "+Turtle.numTanks);
        Turtle.numTanks=5;
        System.out.println("Turtle T1 berusia "+T1.age+" bulan");
        System.out.println("Turtle T1 berada di tangki nomor "+T1.tankNum);
        System.out.println(T1.fishTank());
        System.out.println("");
        System.out.println("Turtle T2 berusia "+T2.age+" bulan");
        System.out.println("Turtle T2 berada di tangki nomor "+T2.tankNum);
        System.out.println(T2.fishTank());
    }
}
```

Penyebab error pada program diatas :

1. **T1.age** seharusnya **T1.getAge()**, kita tidak bisa mengakses class variable secara langsung karena access modifier nya private, dan hanya bisa diakses menggunakan method getter/aksesor yang tersedia di class tersebut dengan nama getAge(). Demikian juga dengan **T2.age**, seharusnya **T2.getAge()**.
2. **T1.tankNum** seharusnya **T1.getTankOfResidence()**, kita tidak bisa mengakses class variable secara langsung karena access modifier nya private, dan hanya bisa diakses menggunakan method getter/aksesor yang tersedia di class tersebut dengan nama getAge(). Demikian juga dengan **T2.tankNum** seharusnya **T2.getTankOfResidence()**.

Turtle.java | DriverTurtle.java

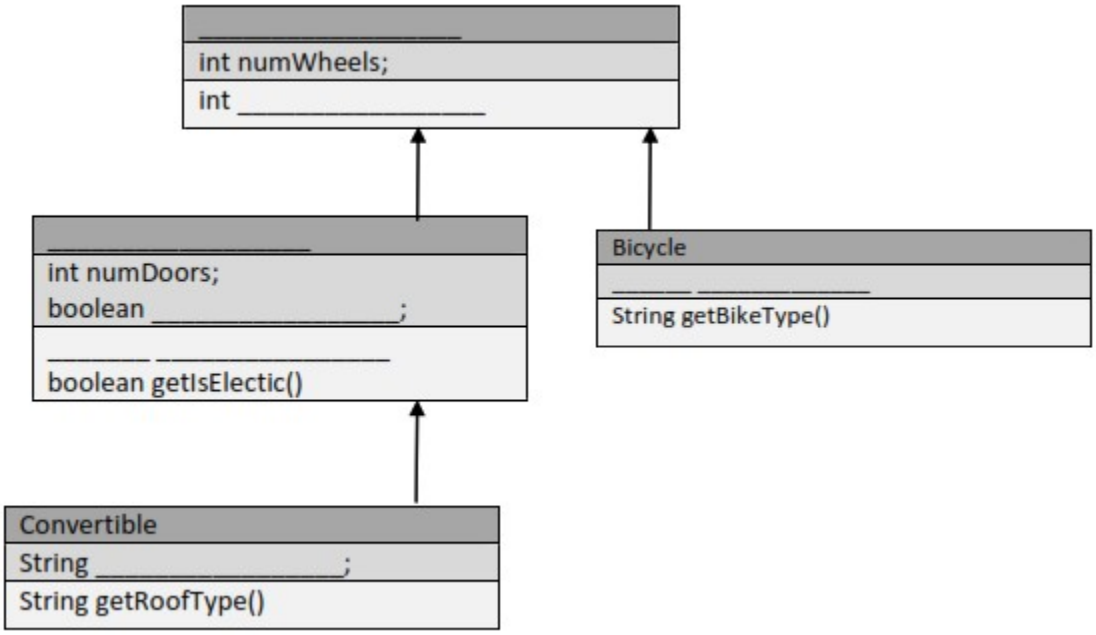
```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class Turtle {
    private static String food = "Turtle feed";
    private static int numTanks = 3;
    private int age, tankNum;

    public Turtle(int age) {
        this.age = age;
        this.tankNum = (int)((Math.random() * numTanks) + 1);
        System.out.printf("Makananku adalah %s\n", Turtle.food);
    }

    public static String fishTanks() {
        return "I have " + numTanks + " fish tanks.";
    }

    /**
     * @return the age
     */
    public int getAge() {
        return this.age;
    }

    /**
     * @return the tankNum
     */
    public int getTankNum() {
        return this.tankNum;
    }

    /**
     * @return the numTanks
     */
    public static int getNumTanks() {
        return numTanks;
    }

    /**
     * @param numTanks the numTanks to set
     */
    public static void setNumTanks(int numTanks) {
        Turtle.numTanks = numTanks;
    }
}
```

Turtle.java | DriverTurtle.java

```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class DriverTurtle {

    public static void main(String[] args) {
        Turtle t1 = new Turtle(1);
        Turtle t2 = new Turtle(2);

        System.out.printf("Jumlah tangki total adalah %s\n\n", Turtle.getNumTanks());
        Turtle.setNumTanks(5);
        System.out.printf("Turtle t1 berusia %d bulan\n", t1.getAge());
        System.out.printf("Turtle t1 berada di tangki nomor %d\n", t1.getTankNum());
        System.out.println(t1.fishTanks());
        System.out.println("-----------------------------------------------------");
        System.out.printf("Turtle t2 berusia %d bulan\n", t2.getAge());
        System.out.printf("Turtle t2 berada di tangki nomor %d\n", t2.getTankNum());
        System.out.println(t2.fishTanks());
    }
}
```

Kode saya diatas adalah kode yang benar. Hanya saja **t1.fishTanks()** dan **t2.fishTanks()** lebih baik diganti dengan **Turtle.fishTanks()**. Karena static variable nilai nya akan tetap sama walau diakses dan dimodifikasi lewat object yang berbeda.

2. Dengan menggunakan kode program dan UML di bawah ini, isi isian dengan keyword yang tepat. Kemudian tulis program dengan Eclipse.




```
public class Vehicle {
    _____ int numWheels;
    _____ Vehicle(int numWheels)
    {
        this.numWheels = numWheels;
    }

    public int getWheels() {
        return wheels;
    }
}

public class Car _____ Vehicle {
    _____ int numDoors;
    _____ boolean isElectric;

    public Car (int numWheels, int numDoors, boolean isElectric) {
        _____(numWheels);
        this.numDoors = numDoors;
        this.isElectric = isElectric;
    }

    public int getNumDoors() {
        return _____;
```

```
    }

    public boolean getIsElectric() {
        return isElectric;
    }
}

public class Bicycle _________ Vehicle {
    //Mountain bike, road bike, recumbent bike.. etc
    _________ String bikeType;

    public Bicycle(int numWheels, String bikeType) {
        super(numWheels);
        this.bikeType = _________;
    }

    public _________ getBikeType() {
        return bikeType;
    }
}

public class Convertible _________ Car {
    //Soft top or rag top, or hard top
    _________ String roofType;

    public Convertible(int numWheels, int numDoors, _________ isElectric, String roofType) {
        super(numWheels, _________, _________);
        this.roofType = roofType;
    }

    _________ String getRoofType() {
        return roofType;
    }
}
```

Vehicle.java | Car.java | Bicycle.java | *Convertible.java

```java
/**
 * @author adam
 */
package com.latihan7;

abstract public class Vehicle {
    private int numWheels;

    public Vehicle(int numWheels) {
        this.numWheels = numWheels;
    }

    /**
     * @return the numWheels
     */
    public int getNumWheels() {
        return this.numWheels;
    }

    /**
     * @param numWheels the numWheels to set
     */
    public void setNumWheels(int numWheels) {
        this.numWheels = numWheels;
    }
}
```

Vehicle.java | Car.java | Bicycle.java | *Conver

```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class Car extends Vehicle{
    private int numDoors;
    private boolean isElectric;

    /**
     * @param numWheels
     */
    public Car(int numWheels, int numDoors, boolean isElectric) {
        super(numWheels);
        this.numDoors = numDoors;
        this.isElectric = isElectric;
    }

    /**
     * @return the numDoors
     */
    public int getNumDoors() {
        return this.numDoors;
    }

    /**
     * @param numDoors the numDoors to set
     */
    public void setNumDoors(int numDoors) {
        this.numDoors = numDoors;
    }

    /**
     * @return the isElectric
     */
    public boolean isElectric() {
        return this.isElectric;
    }

    /**
     * @param isElectric the isElectric to set
     */
    public void setElectric(boolean isElectric) {
        this.isElectric = isElectric;
    }
}
```

Vehicle.java | Car.java | Bicycle.java | *Convertible.java

```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class Bicycle extends Vehicle {
    private String bikeType;

    /**
     * @param numWheels
     */
    public Bicycle(int numWheels, String bikeType) {
        super(numWheels);
        this.bikeType = bikeType;
    }

    /**
     * @return the bikeType
     */
    public String getBikeType() {
        return this.bikeType;
    }

    /**
     * @param bikeType the bikeType to set
     */
    public void setBikeType(String bikeType) {
        this.bikeType = bikeType;
    }
}
```

Vehicle.java | Car.java | Bicycle.java | Convertible.java

```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class Convertible extends Car {
    private String roofType;

    /**
     * @param numWheels
     */
    public Convertible(int numWheels, int numDoors, boolean isElectric, String roofType) {
        super(numWheels, numDoors, isElectric);
        this.roofType = roofType;
    }

    /**
     * @return the roofType
     */
    public String getRoofType() {
        return this.roofType;
    }

    /**
     * @param roofType the roofType to set
     */
    public void setRoofType(String roofType) {
        this.roofType = roofType;
    }
}
```

3. Tulis program berikut kemudian jalankan! Beri penjelasan mengenai output yang didapatkan!

```
public class A {
    void callthis() {
        System.out.println("Inside Class A's Method!");
        }
}


public class B extends A{
    void callthis() {
         System.out.println("Inside Class B's Method!");
         }
}


public class C extends A {
    void callthis() {
        System.out.println("Inside Class C's Method!");
        }
}


public class DynamicDispatch {
    public static void main(String args[])  {
        A a = new A();
        B b = new B();
        C c = new C();
        A ref;
        ref = b;
        ref.callthis();
        ref = c;
        ref.callthis();
        ref = a;
        ref.callthis();
        }
}
```

A.java | B.java | C.java | DynamicDispatch.java

```
1  /**
4  package com.latihan7;
5
7   * @author adam
10 public class A {
11
12     public void callThis() {
13         System.out.println("Ini didalam methodnya kelas A!");
14     }
15 }
```

A.java | B.java | C.java | DynamicDispatch.java

```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class B extends A{

    @Override
    public void callThis() {
        System.out.println("Ini didalam methodnya kelas B!");
    }
}
```

A.java | B.java | C.java | DynamicDispatch.java

```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class C extends A {

    @Override
    public void callThis() {
        System.out.println("Ini didalam methodnya kelas C!");
    }
}
```

Console

```
<terminated> DynamicDispatch [Java Application] /usr/lib/jvm/java-13-op
Ini didalam methodnya kelas B!
Ini didalam methodnya kelas C!
Ini didalam methodnya kelas A!
```

LATIHAN:

4. Diberikan class Shape, Rectangle, Circle, Triangle, Square dan Object. Gambarkan diagram UML serta tuliskan kode programnya dengan definisi sbb:
   - Square adalah sebuah Rectangle
   - Rectangle, Triangle dan Circle merupakan tipe Shape
   - Shape memiliki field color bertipe String dan method getColor() serta method hitungLuas().
   - Rectangle memiliki field tambahan yakni panjang dan lebar bertipe int.
   - Circle memiliki field tambahan yakni jejari.
   - Triangle memiliki field tambahan alas dan tinggi.
   - Rectangle, Circle dan Triangle memiliki method setter dan getter terhadap semua field.
   - Rectangle, Circle dan Triangle melakukan override terhadap hitungLuas() dengan definisi luas menyesuaikan bentuknya.
   - Square memiliki method setSisi() untuk mengatur nilai panjang dan lebar dengan isian yang sama.

Shape.java | Rectangle.java | Circle.java | Triangle.java

```
1  /**
4  package com.latihan7;
6   * @author adam
9  abstract public class Shape {
10     private String color;
11
12     public Shape(String color) {
13         this.color = color;
14     }
15
17      * @return the color
19     public String getColor() {
20         return this.color;
21     }
22
24      * @param color the color to set
26     public void setColor(String color) {
27         this.color = color;
28     }
29
30     abstract  public int hitungLuas();
31 }
```

Shape.java | *Rectangle.java | Circle.java

```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class Rectangle extends Shape{
    private int panjang, lebar;

    /**
     * @param color
     */
    public Rectangle(String color, int panjang, int lebar) {
        super(color);
        this.panjang = panjang;
        this.lebar = lebar;
    }

    @Override
    public int hitungLuas() {
        return this.panjang * this.lebar;
    }

    /**
     * @return the panjang
     */
    public int getPanjang() {
        return this.panjang;
    }

    /**
     * @param panjang the panjang to set
     */
    public void setPanjang(int panjang) {
        this.panjang = panjang;
    }

    /**
     * @return the lebar
     */
    public int getLebar() {
        return this.lebar;
    }

    /**
     * @param lebar the lebar to set
     */
    public void setLebar(int lebar) {
        this.lebar = lebar;
    }
}
```

Shape.java | Rectangle.java | *Circle.java | Triangle.java

```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class Circle extends Shape {
    private int jejari;

    /**
     * @param color
     */
    public Circle(String color, int jejari) {
        super(color);
        this.jejari = jejari;
    }

    @Override
    public int hitungLuas() {
        return (int) (3.14 * Math.pow(jejari, 2));
    }

    /**
     * @return the jejari
     */
    public int getJejari() {
        return this.jejari;
    }

    /**
     * @param jejari the jejari to set
     */
    public void setJejari(int jejari) {
        this.jejari = jejari;
    }
}
```

Shape.java | Rectangle.java | Circle.java

```java
/**
 */
package com.latihan7;

/**
 * @author adam
 */
public class Triangle extends Shape {
    private int alas, tinggi;

    /**
     * @param color
     */
    public Triangle(String color, int alas, int tinggi) {
        super(color);
        this.alas = alas;
        this.tinggi = tinggi;
    }

    @Override
    public int hitungLuas() {
        return (int)0.5 * alas * tinggi;
    }

    /**
     * @return the alas
     */
    public int getAlas() {
        return this.alas;
    }

    /**
     * @param alas the alas to set
     */
    public void setAlas(int alas) {
        this.alas = alas;
    }

    /**
     * @return the tinggi
     */
    public int getTinggi() {
        return this.tinggi;
    }

    /**
     * @param tinggi the tinggi to set
     */
    public void setTinggi(int tinggi) {
        this.tinggi = tinggi;
    }
}
```

Shape.java | Rectangle.java | Circle.java | Triangle.java

```java
/**
 */
package com.latihan7;

/**
 *
 * @author adam
 */
public class Square extends Rectangle {
    private int sisi;

    /**
     * @param color
     */
    public Square(String color, int sisi) {
        super(color, sisi, sisi);
    }

    @Override
    public int hitungLuas() {
        return (int) Math.pow(sisi, 2);
    }

    /**
     * @return the sisi
     */
    public int getSisi() {
        return this.sisi;
    }

    /**
     * @param sisi the sisi to set
     */
    public void setSisi(int sisi) {
        this.sisi = sisi;
    }
}
```

5. Berdasarkan kode program poin 4, tuliskan kelas driver untuk membuat objek dan menerapkan konsep polimorfisme.

Setelah sesi praktikum SELESAI, laporan praktikum harus dikirim/diupload ke google classroom pada hari yang ditentukan.