

Object Oriented Programming

Class Relation



Name

Muhammad Baihaqi Aulia Asy'ari

NIM

2241720145

Class

2I

Department

Information Technology

Study Program

D4 Informatics Engineering

Contents

1	Experiment 1	2
1.1	Question	5
1.2	Answer	6
2	Experiment 2	7
2.1	Question	11
2.2	Answer	11
3	Experiment 3	12
3.1	Question	15
3.2	Answer	15

1 Experiment 1

Processor.java

```
1 package classrelationship.experiment1;
2
3 public class Processor {
4     private String brand;
5     private double cache;
6
7     public Processor() {
8     }
9
10    public Processor(String brand, double cache) {
11        this.brand = brand;
12        this.cache = cache;
13    }
14
15    public void setBrand(String brand) {
16        this.brand = brand;
17    }
18
19    public String getBrand() {
20        return brand;
21    }
22
23    public void setCache(double cache) {
24        this.cache = cache;
25    }
26
27    public double getCache() {
28        return cache;
29    }
30
31    public void info() {
32        System.out.printf("Merk Processor = %s\n", brand);
33        System.out.printf("Cache Memory = %.2f\n", cache);
34    }
35 }
```

Laptop.java

```
1 package classrelationship.experiment1;
2
3 public class Laptop {
4     private String brand;
5     private Processor processor;
6
7     public Laptop() {
8     }
9
10    public Laptop(String brand, Processor processor) {
11        this.brand = brand;
12        this.processor = processor;
13    }
14
15    public void setBrand(String brand) {
16        this.brand = brand;
17    }
18
19    public void setProcessor(Processor processor) {
20        this.processor = processor;
21    }
22
23    public void info() {
24        System.out.println("Merk Laptop = " + brand);
25        processor.info();
26    }
27 }
```

MainExperiment1.java

```
1 package classrelationship.experiment1;
2
3 public class MainExperiment1 {
4     public static void main(String[] args) {
5         Processor processor = new Processor("Intel i5", 3);
6         Laptop laptop = new Laptop("Thinkpad", processor);
7
8         laptop.info();
9
10        Processor p1 = new Processor();
11        p1.setBrand("Intel i5");
12        p1.setCache(4);
13        Laptop l1 = new Laptop();
14        l1.setBrand("Thinkpad");
15        l1.setProcessor(p1);
16        l1.info();
17    }
18 }
```

Terminal

```
1 PS D:\Kuliah> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe'
   ↳ '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
   ↳ 'C:\Users\G4CE-PC\AppData\Roaming\Code\User\workspaceStorage\
   ↳ 80d97a47d24665dc0bce7ab1e048ecbd\redhat.java\jdt_ws\
   ↳ Kuliah_28156aa7\bin'
   ↳ 'classrelationship.experiment1.MainExperiment1'
2 Merk Laptop = Thinkpad
3 Merk Processor = Intel i5
4 Cache Memory = 3.00
5 Merk Laptop = Thinkpad
6 Merk Processor = Intel i5
7 Cache Memory = 4.00
```

1.1 Question

Berdasarkan percobaan 1, jawablah pertanyaan-pertanyaan yang terkait:

1. Di dalam *class Processor* dan *class Laptop* , terdapat method *setter* dan *getter* untuk masing-masing atributnya. Apakah gunanya *method setter* dan *getter* tersebut ?
2. Di dalam *class Processor* dan *class Laptop*, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?
3. Perhatikan *class Laptop*, di antara 2 atribut yang dimiliki (*merk* dan *proc*), atribut manakah yang bertipe *object* ?
4. Perhatikan *class Laptop*, pada baris manakah yang menunjukkan bahwa *class Laptop* memiliki relasi dengan *class Processor* ?
5. Perhatikan pada *class Laptop* , Apakah guna dari sintaks `proc.info()` ?
6. Pada *class MainPercobaan1*, terdapat baris kode:

```
Laptop l = new Laptop("Thinkpad", p);
```

Apakah p tersebut ?

Dan apakah yang terjadi jika baris kode tersebut diubah menjadi:

```
Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));
```

Bagaimanakah hasil program saat dijalankan, apakah ada perubahan ?

1.2 Answer

1. To set or get a value that has to follow a certain rule (encapsulation) and to set a value when the default constructor is used.
2. Default constructor makes us declare the value of the object's attribute using the setter. The parametric constructor set the value accordingly in the instantiation of the object.
3. proc is the object attribute. The attribute derived from the Processor class.
4. The line where it state the attribute of class Laptop uses the Processor class as an object attribute. In my instance is in line 5 of the class Laptop.java .
5. It is used to call the method info() in the Laptop class which give description on the instance of said Laptop object.
6. The p is variable used to store the instance of the Processor object that has been instantiated. Nothing will change in the eyes of the users if that method of instantiation is used.

2 Experiment 2

Car.java

```
1 package classrelationship.experiment2;
2
3 public class Car {
4     private String brand;
5     private int cost;
6
7     public Car() {
8     }
9
10    public void setBrand(String brand) {
11        this.brand = brand;
12    }
13
14    public String getBrand() {
15        return brand;
16    }
17
18    public void setCost(int cost) {
19        this.cost = cost;
20    }
21
22    public int getCost() {
23        return cost;
24    }
25
26    public int calculateCarCost(int day) {
27        return cost * day;
28    }
29 }
```

Driver.java

```
1 package classrelationship.experiment2;
2
3 public class Driver {
4     private String name;
5     private int cost;
6
7     public Driver() {
8     }
9
10    public void setName(String name) {
11        this.name = name;
12    }
13
14    public String getName() {
15        return name;
16    }
17
18    public void setCost(int cost) {
19        this.cost = cost;
20    }
21
22    public int getCost() {
23        return cost;
24    }
25
26    public int calculateDriverCost(int day) {
27        return cost * day;
28    }
29 }
```

User.java

```
1 package classrelationship.experiment2;
2
3 public class User {
4     private String name;
5     private Car car;
6     private Driver driver;
7     private int day;
8
9     public User() {
10    }
11
12    public void setName(String name) {
13        this.name = name;
14    }
15
16    public String getName() {
17        return name;
18    }
19
20    public void setCar(Car car) {
21        this.car = car;
22    }
23
24    public Car getCar() {
25        return car;
26    }
27
28    public void setDriver(Driver driver) {
29        this.driver = driver;
30    }
31
32    public Driver getDriver() {
33        return driver;
34    }
35
36    public void setDay(int day) {
37        this.day = day;
38    }
39
40    public int getDay() {
41        return day;
```

```

42     }
43
44     public int calculateTotalCost() {
45         return car.calculateCarCost(day) +
46             ↪ driver.calculateDriverCost(day);
47     }
48 }

```

MainExperiment2.java

```

1  package classrelationship.experiment2;
2
3  public class MainExperiment2 {
4      public static void main(String[] args) {
5          Car car = new Car();
6          car.setBrand("Avanza");
7          car.setCost(350_000);
8
9          Driver driver = new Driver();
10         driver.setName("John Doe");
11         driver.setCost(200_000);
12
13         User user = new User();
14         user.setName("Jane Doe");
15         user.setCar(car);
16         user.setDriver(driver);
17         user.setDay(2);
18
19         System.out.println("Total Cost = " +
20             ↪ user.calculateTotalCost());
21     }
22 }

```

Terminal

```

1  PS D:\Kuliah> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe'
   ↪ '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
   ↪ 'C:\Users\G4CE-PC\AppData\Roaming\Code\User\workspaceStorage\
   ↪ 80d97a47d24665dc0bce7ab1e048ecbd\redhat.java\jdt_ws\
   ↪ Kuliah_28156aa7\bin'
   ↪ 'classrelationship.experiment2.MainExperiment2'
2  Total Cost = 1100000

```

2.1 Question

1. Perhatikan class Pelanggan. Pada baris program manakah yang menunjukkan bahwa class Pelanggan memiliki relasi dengan class Mobil dan class Sopir ?
2. Perhatikan method hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda method tersebut harus memiliki argument hari ?
3. Perhatikan kode dari class Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?
4. Perhatikan class MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s) ?
5. Perhatikan class MainPercobaan2. Untuk apakah proses p.hitungBiayaTotal() tersebut ?
6. Perhatikan class MainPercobaan2, coba tambahkan pada baris terakhir dari method main dan amati perubahan saat di-run!

```
System.out.println(p.getMobil().getMerk());
```

Jadi untuk apakah sintaks p.getMobil().getMerk() yang ada di dalam method main tersebut?

2.2 Answer

1. On the line where it's declaring the attribute as an object of Car and Driver. In this case, line 5 and 6.
2. Because both class don't have and don't know how many day it will be. The day attribute is owned by the User class.
3. To get the calculation on the cost of the car rent and the driver fee.
4. To set the User attribute using the instantiated car object and driver object.
5. To get the sum of all cost for the user.
6. To get the brand name of the car used by the user.

3 Experiment 3

Employee.java

```
1 package classrelationship.experiment3;
2
3 public class Employee {
4     private String nip;
5     private String name;
6
7     public Employee(String nip, String name) {
8         this.nip = nip;
9         this.name = name;
10    }
11
12    public void setNip(String nip) {
13        this.nip = nip;
14    }
15
16    public String getNip() {
17        return nip;
18    }
19
20    public void setName(String name) {
21        this.name = name;
22    }
23
24    public String getName() {
25        return name;
26    }
27
28    public String info() {
29        String info = "";
30        info += "NIP: " + this.nip + "\n";
31        info += "Name: " + this.name + "\n";
32        return info;
33    }
34 }
```

Train.java

```
1 package classrelationship.experiment3;
2
3 public class Train {
4     private String name;
5     private String classification;
6     private Employee conductor;
7     private Employee assitant;
8
9     public Train(String name, String classification, Employee
    ↪ conductor) {
10         this.name = name;
11         this.classification = classification;
12         this.conductor = conductor;
13     }
14
15     public Train(String name, String classification, Employee
    ↪ conductor, Employee assistant) {
16         this.name = name;
17         this.classification = classification;
18         this.conductor = conductor;
19         this.assitant = assistant;
20     }
21
22     public void setName(String name) {
23         this.name = name;
24     }
25
26     public String getName() {
27         return name;
28     }
29
30     public void setClassification(String classification) {
31         this.classification = classification;
32     }
33
34     public String getClassification() {
35         return classification;
36     }
37
38     public void setConductor(Employee conductor) {
39         this.conductor = conductor;
```

```

40     }
41
42     public Employee getConductor() {
43         return conductor;
44     }
45
46     public void setAssitant(Employee assitant) {
47         this.assitant = assitant;
48     }
49
50     public Employee getAssitant() {
51         return assitant;
52     }
53
54     public String info() {
55         String info = "";
56         info += "Name      : " + this.name + "\n";
57         info += "Class      : " + this.classification + "\n";
58         info += "Conductor: " + this.conductor.info() + "\n";
59         info += "Assistant: " + this.assitant.info() + "\n";
60         return info;
61     }
62 }

```

MainExperiment3.java

```

1  package classrelationship.experiment3;
2
3  public class MainExperiment3 {
4      public static void main(String[] args) {
5          Employee conductor = new Employee("1234", "Spongebob
6              ↪ Squarepants");
7          Employee assistant = new Employee("4567", "Patrick Star");
8          Train train = new Train("New Style", "Bussiness", conductor,
9              ↪ assistant);
10         System.out.println(train.info());
11     }
12 }

```

Terminal

```
1 PS D:\Kuliah> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe'
   ↪ '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
   ↪ 'C:\Users\G4CE-PC\AppData\Roaming\Code\User\workspaceStorage\
   ↪ 80d97a47d24665dc0bce7ab1e048ecbd\redhat.java\jdt_ws\
   ↪ Kuliah_28156aa7\bin'
   ↪ 'classrelationship.experiment3.MainExperiment3'
2 Name      : New Style
3 Class     : Bussiness
4 Conductor: NIP: 1234
5 Name: Spongebob Squarepants
6
7 Assistant: NIP: 4567
8 Name: Patrick Star
```

3.1 Question

1. Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa?
2. Buatlah main program baru dengan nama class MainPertanyaan pada package yang sama. Tambahkan kode berikut pada method main() !

```
Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis",
   ↪ masinis);
System.out.println(keretaApi.info());
```

3. Apa hasil output dari main program tersebut ? Mengapa hal tersebut dapat terjadi ?
4. Perbaiki class KeretaApi sehingga program dapat berjalan !

3.2 Answer

1. To get the info of the object in a form of a String to be append in the info String of the Train info method.
2. MainQuestion.java

```
1 package classrelationship.experiment3;
2
3 public class MainQuestion {
4     public static void main(String[] args) {
```

```
5         Employee conductor = new Employee("1234", "Spongebob
        ↳ Squarepants");
6         Train train = new Train("New Style", "Bussiness",
        ↳ conductor);
7         System.out.println(train.info());
8     }
9 }
```

3. Terminal

```
1 PS D:\Kuliah> & 'C:\Program
    ↳ Files\Java\jdk-18.0.2.1\bin\java.exe'
    ↳ '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
    ↳ 'C:\Users\G4CE-PC\AppData\Roaming\Code\User\
    ↳ workspaceStorage\80d97a47d24665dc0bce7ab1e048ecbd\
    ↳ redhat.java\jdt_ws\Kuliah_28156aa7\bin'
    ↳ 'classrelationship.experiment3.MainQuestion'
2 Exception in thread "main" java.lang.NullPointerException: Cannot
    ↳ invoke "classrelationship.experiment3.Employee.info()"
    ↳ because "this.assitant" is null
3     at classrelationship.experiment3.Train.info (Train.java:59)
4     at classrelationship.experiment3.MainQuestion.main
    ↳ (MainQuestion.java:7)
```

Because on the Train info() method it also ask for the data of the assistant which is nonexistence in this instance.