

Data Structure and Algorithm Practicum

Class and Object



Name

Muhammad Baihaqi Aulia Asy'ari

NIM

2241720145

Class

1I

Department

Information Technology

Study Program

D4 Informatics Engineering

11 Question

1. Class/Object has a characteristic of having attributes and methods.
2. To declare a class, the key word to be use is `class`.
3. There are 4 total attributes, 2 of them are string and the other are integer. the attributes name are `namaBarang`, `jenisBarang`, `stok`, and `hargaSatuan`.
4. The attributes were declared on line 4 and line 5.
5. There are 4 methods inside the class 3 of them are void methods and one of them is string method.
6. The methods were declared on line 7 until line 22.
7.

```
void kurangiStok(int n) {  
    if (stok <= 0) {  
        stok = stok - n;  
    }  
}
```
8. The integer parameter is a variable to be used in the mathematical operation in the function and must be in a form of integer to perform the mathematical operation.
9. Because the function of `hitungHargaTotal()` has an output from the calculation of total price in a form of integer.
10. Because `tambahStok()` only change the value of a variable and does not need to output a value.
11. The instantiation process happend at line 5 and the object created is named `b1`
12. To access the attributes and methods of the object `b1`, write `b1.methods()` or `b1.attributes`
13. The parametric constructor in the class `Barang` in part 4 was declared in line 9 until line 14
14. In line 16 of class `BarangMain` in part 4, what the code is doing is instantiate an object called `b2` using the parametric constructor
15.

```
Barang b3 = new Barang("KTT Kang White", "Keyboard Switch", 9999,  
    ↪ 3_000);
```

12 Task

```
1. public class Lingkaran {
    double phi = Math.PI, r;
    Lingkaran() {

    }
    Lingkaran(double r) {
        this.r = r;
    }
    double hitungLuas() {
        return phi * r * r;
    }
    double hitungKeliling() {
        return 2 * r * phi;
    }
}
```

2.

```
public class RentalTransaction {
    String memberId, memberName, gameName;
    int dailyPrice, dayRent;
    RentalTransaction() {

    }
    RentalTransaction(String memberId, String memberName, String
    ↪ gameName, int dailyPrice, int dayRent) {
        this.memberId = memberId;
        this.memberName = memberName;
        this.gameName = gameName;
        this.dailyPrice = dailyPrice;
        this.dayRent = dayRent;
    }
    void print() {
        System.out.printf("""
            Member ID    : %s
            Member name  : %s
            Game name    : %s
            Price        : %d
            Rent period  : %d %s
            Total        : %d
        """)
    }
```

```

        """, memberId, memberName, gameName, dailyPrice,
        ↪ dayRent, dayRent < 2 ? "day" : "days",
        ↪ dailyPrice * dayRent);
    }
}
-

```

RentalTransaction
memberId : String memberName : String gameName : String dailyPrice : 'int dayRent : int
print() : void

```

3. public class Item {
    String name;
    int unitPrice, qty;
    Item() {

    }
    Item(String name, int unitPrice, int qty) {
        this.name = name;
        this.unitPrice = unitPrice;
        this.qty = qty;
    }
    int calculateTotalPrice() {
        return qty * unitPrice;
    }
    int calculateDiscount() {
        int totalPrice = calculateTotalPrice();
        if (totalPrice > 100_000) return (int) 0.9;
        if (totalPrice > 50_000) return (int) 0.95;
        return 1;
    }
    int calculateFinalPrice() {
        int totalPrice = calculateTotalPrice();
        int discount = calculateDiscount();
        return totalPrice * discount;
    }
}

```

```
4. public class PacMan {
    int x, y, height, width;
    PacMan() {

    }
    PacMan(int x, int y, int height, int width) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
    }
    void moveLeft() {
        if (x > 0) x--;
    }
    void moveRight() {
        if (x < width) x++;
    }
    void moveDown() {
        if (y > 0) y--;
    }
    void moveUp() {
        if (y < height) y++;
    }
    void printCoordinate() {
        System.out.printf("x: %d y: %d", x, y);
    }
}
```