OBJECT ORIENTED PROGRAMMING

[QUIZ 1]

AYU JOVITA WIDYADHARI
2241720219/04/2I

QUIZ 1

1. Class and Object:

    - What is a "class" in object-oriented programming?

      Answer:

      In object-oriented programming, a class can be referred to as a blueprint or template for creating objects (instances). A class contains a set of attributes (data members) and methods (functions) that are common to all objects in the class. Classes encapsulate the behavior and data of objects.

    - How do you define the object of a class in the Java programming language?

      Answer:

      The way to define an object of a class in Java is by using the keyword **'new'** followed by the class name and parentheses.

      For example: if you have a class called Item, you can create an object of that class like this: **Item laptop = new Item();**

    - Suppose you have an "Item" class in an inventory information system.How would you create a "laptop" object from that class?

      Answer:

      Suppose you have an "Item" class in an inventory information system. How would you create a "laptop" object from that class?

      Item laptop = new Item();

2. Encapsulation:

    - Explain the concept of encapsulation in object-oriented programming and why it is important in the development of an item inventory information system.

      Answer:

      Encapsulation is the bundling of data (attributes) and the methods (functions) that operate on that data into a single unit known as a class. It restricts direct access to some of an object's components and prevents the accidental modification of data. Encapsulation helps in data hiding, reducing complexity, and promoting modularity and reusability of code, making it crucial in the development of an item inventory information system to maintain data integrity and security.

    - In the context of an inventory information system, name an example of an attribute (variable) that should be encapsulated and why.

      Answer:

      An example of an encapsulated attribute in an inventory system could be the item's price. By encapsulating the price attribute, you can control how it is accessed and modified, ensuring that it follows specific rules or validations, such as not allowing a negative price or limiting the range of acceptable values.

3. Class Relation:

   - What is the relationship between classes in object-oriented programming? programming?

     Answer:

     In object-oriented programming, classes can have various relationships, including inheritance (where one class inherits properties and methods from another), association (where classes are related but not dependent on each other), aggregation (a type of association where one class contains objects of another class), and composition (a stronger form of aggregation where the child object cannot exist without the parent object).

   - In an item inventory information system, how would you describe the the relationship between class "Item" and class "Category"?

     Answer:

     The relationship between the "Item" class and the "Category" class could be an association. An item is associated with a category, meaning each item belongs to a specific category. This association allows items to be categorized and organized efficiently within the inventory system.

4. PBL:

   - Based on the case of the item inventory information system, try to create a simple class with attributes and methods that describe an entity. And its attributes and methods that describe an entity in the system (for example, the class "Goods"). In the system (for example, the class "Goods").

     ```
     public class Goods {
       private String itemName;
       private double price;
       private int quantity;

       public Goods(String itemName, double price, int quantity) {
         this.itemName = itemName;
         this.price = price;
         this.quantity = quantity;
       }

       public String getItemName() {
         return itemName;
     ```

```
    }

    public double getPrice() {
        return price;
    }

    public int getQuantity() {
        return quantity;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    public double calculateTotalValue() {
        return price * quantity;
    }
}
```

- How will you use encapsulation to protect the attributes in the class?

  In the Goods class, the attributes item name, price, and quantity are declared as private. This means they can only be accessed within the class itself. Access to these attributes is provided through public methods (getItemName(), getPrice(), getQuantity(), setPrice(), and setQuantity()) that encapsulate the data. This encapsulation is useful for ensuring that attributes can only be modified in a controlled manner, maintaining data integrity.

- Describe the class hierarchy or the relationship between classes that may exist in the system. Information system in the Information Technology department. Give an example of relationships between classes (e.g., inheritance or association) in that context.

In an inventory information system, there might be a class hierarchy where more specific classes inherit from more general ones. For instance, we could have a general Item class, and then more specific classes like ElectronicsItem and ClothingItem that inherit from the Item class. This demonstrates inheritance, where the specialized classes inherit attributes and methods from the general class.

Additionally, there could be an association between classes. For example, a Supplier class could be associated with the Item class. Each item might have a supplier, establishing a relationship between the Item class and the Supplier class. This association allows the system to track which supplier provides each item in the inventory.