FPT SOFTWARE

FRESHER ACADEMY

BASIC JAVA

Ngattt (FHO.FA)

04/20/2022

[Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.]

Contents

Software	**Assignment topic** : Java Basic Lab **Assignment duration** : 60 minutes	**FRESHER ACADEMY**

Objective

- Fresher can apply knowledge about OOP to code a simple application.
- Employee Management with Different Department: Suppose you are building an employee management system that has different departments such as IT, HR, Sales and Marketing.
    - Create a abstract class called **Employee** with instance variables for *name*, *id*, and *department*. Then create subclasses called **ITEmployee** with *level* attribute, **HREmployee** with *yearsOfExperience* attribute, SalesEmployee with *salesTarget* attribute and **MarketingEmployee** with *product* attribute.
    - Implement appropriate constructors, getters, and setters for all classes.
    - Implement a method called assignTask() in the Employee class that assigns a task to the employee based on their department. The ITEmployee class should be assigned programming tasks, the HREmployee class should be assigned employee management tasks, and the MarketingEmployee class should be assigned marketing tasks.
    - Implement *debugCode*() for **ITEmployee**, *conductInterview*() for HREmployee, *createCampaign*() for **MarketingEmployee**
- Fresher can create abstract class, subclasses and write some methods
- Fresher can create some instances of class and test all the methods of the class.

Business needs

- Users can create as many instances of Employee, ITEmployee, HREmployee, MarketingEmployee as they want. After tests all the methods of the instances, the program will print like this:

```
*** 1. Sales Employee
- Name: Sarah
- ID: 1
- Sales Target: 5000.0
- Tasks:
        + Task assigned: Make a sales presentation
        + Task assigned: Follow up with potential client
        + Task assigned: Prepare a presentation for a meeting with a client
- Sarah has reached the sales target!
*** 2. IT Employee
- Name: Alice
- Department: IT
- Level: Senior
- Tasks:
        + Programming task: Fix a bug in the system
        + Programming task: Optimize the website for faster loading
        + Programming task: Develop a new feature for the mobile app
- Debugging code for Alice in IT department.
```

```
*** 3. HREmployee
- Name: Bob
- Department: HR
- Years Of Experience: 5
- Tasks:
        + Employee management task: Interview a candidate
        + Employee management task: Review employee benefits package
        + Employee management task: Conduct a performance review
- Conducting interview for Bob in HR department.
*** 4. MarketingEmployee
- Name: Emily
- Department: Marketing
- Product: Product X
- Tasks:
        + Marketing task: Create a social media campaign
        + Marketing task: Design a product brochure
- Creating marketing campaign for Emily in Marketing department.
```

- Do not verify information user input

Working requirements

- Working environment: Eclipse/IntelliJ/NetBeans.

Technologies

- The product implements Java program language with: Abstract class, Subclasses

Project Descriptions

1. **Step 1: Create a new Java project** (skip this step if you already have **YourFullName_JavaSE** project)
   - Open your preferred Java IDE (such as Eclipse, IntelliJ, or NetBeans).
   - Create a new Java project by selecting File > New > Java Project.
   - Name the project **YourFullName_JavaSE, for example "NguyenVanA_JavaSE"** and click Finish.
2. **Step 2: Create a package** (skip this step if you already have **lab4** package)
   - In the project explorer, right-click on the src folder and select New > Package.
   - Name the package "lab4" and click Finish.
3. **Step 3: Create an abstract class**
   - In the Package Explorer panel on the left side of the screen, right-click on the '**lab4**' package to create the class.
   - Select "New" from the context menu, then "Class" from the submenu.
   - In the "New Java Class" dialog box, enter "**Employee**" for the class in the "Name" field.
   - Click checkbox abstract.
   - Click "Finish" to create the class.
4. **Step 4: Create a class**

- In the Package Explorer panel on the left side of the screen, right-click on the '**lab4**' package to create the class.
- Select "New" from the context menu, then "Class" from the submenu.
- In the "New Java Class" dialog box, enter "**SalesEmployee**" for the class in the "Name" field.
- Click "Finish" to create the class.

5. **Step 5: Create a class**
   - In the Package Explorer panel on the left side of the screen, right-click on the '**lab4**' package to create the class.
   - Select "New" from the context menu, then "Class" from the submenu.
   - In the "New Java Class" dialog box, enter "**ITEmployee**" for the class in the "Name" field.
   - Click "Finish" to create the class.
6. **Step 6: Create a class**
   - In the Package Explorer panel on the left side of the screen, right-click on the '**lab4**' package to create the class.
   - Select "New" from the context menu, then "Class" from the submenu.
   - In the "New Java Class" dialog box, enter "**HREmployee**" for the class in the "Name" field.
   - Click "Finish" to create the class.
7. **Step 7: Create a class**
   - In the Package Explorer panel on the left side of the screen, right-click on the '**lab4**' package to create the class.
   - Select "New" from the context menu, then "Class" from the submenu.
   - In the "New Java Class" dialog box, enter "**MarketingEmployee**" for the class in the "Name" field.
   - Click "Finish" to create the class.
8. **Step 8: Create a class for testing**
   - In the Package Explorer panel on the left side of the screen, right-click on the '**lab4**' package to create the class.
   - Select "New" from the context menu, then "Class" from the submenu.
   - In the "New Java Class" dialog box, enter "**EmployeeTest**" for the class in the "Name" field.
   - Click "Finish" to create the class.
9. **Step 9: Coding attributes and methods for Employee (abstract class)**
   - Attributes:

```
protected String name;
protected int id;
protected String department;
```

- Methods:

```
public void assignTask(String task) {
    System.out.println("\t+ Task assigned: " + task);
}
```

10.Step 10: Coding attributes and methods for ITEmployee

- Attributes: private String **level;** //level is Senior / Fresher
- Methods: assignTask và debugCode

```
public void assignTask(String task) {
    System.out.println("\t+Programming task: " + task);
}

public void debugCode() {
    System.out.println("- Debugging code for " + name + " in IT department.");
}
```

11. Step 11: Coding attributes and methods for HREmployee class

- Attributes: private int yearsOfExperience;
- Methods: **assignTask** and **conductInterview**

```
public void assignTask(String task) {
    System.out.println("\t+ Employee management task: " + task);
}

public void conductInterview() {
    System.out.println("- Conducting interview for " + name
            + " in HR department.");
}
```

12. Step 12: Coding attributes and methods for MarketingEmployee

- Attributes: private String **level;**
- Methods: **assignTask** and **createCampaign**

```
public void assignTask(String task) {
    System.out.println("\t+ Marketing task: " + task);
}

public void createCampaign() {
    System.out.println("- Creating marketing campaign for " + name
            + " in Marketing department.");
}
```

13. Step 13: Coding attributes and methods for SalesEmployee

- Attributes: private double salesTarget;
- Methods: **makeSale**

```
    public void makeSale(double amount) {
        if (amount >= salesTarget) {
            System.out.println("- " + getName() + " has reached the sales target!");
        } else {
            System.out.println("- " + getName() + " has not reached the sales target.");
        }
    }
```

14. Step 14: Coding main() method for EmployeeTest class

Write main() method to create some instances of BankAccount and then test all the methods:

```
    SalesEmployee salesEmployee1 = new SalesEmployee("Sarah", 1, 5000.0);
    System.out.println("*** 1. Sales Employee");
    System.out.println("- Name: " + salesEmployee1.getName());
    System.out.println("- ID: " + salesEmployee1.getId());
    System.out.println("- Sales Target: " + salesEmployee1.getSalesTarget());
    System.out.println("- Tasks: ");
    salesEmployee1.assignTask("Make a sales presentation");
    salesEmployee1.assignTask("Follow up with potential client");
    salesEmployee1.assignTask("Prepare a presentation for a meeting with a client");
    salesEmployee1.makeSale(5000.0);

    System.out.println("*** 2. IT Employee");
    ITEmployee itEmployee1 = new ITEmployee("Alice", 2, "Senior", "IT");
    System.out.println("- Name: " + itEmployee1.getName());
    System.out.println("- Department: " + itEmployee1.getDepartment());
    System.out.println("- Level: " + itEmployee1.getLevel());
    System.out.println("- Tasks: ");
    itEmployee1.assignTask("Fix a bug in the system");
    itEmployee1.assignTask("Optimize the website for faster loading");
    itEmployee1.assignTask("Develop a new feature for the mobile app");
    itEmployee1.debugCode();

    HREmployee hrEmployee1 = new HREmployee("Bob", 3, 5);
    System.out.println("*** 3. HREmployee");
    System.out.println("- Name: " + hrEmployee1.getName());
    System.out.println("- Department: " + hrEmployee1.getDepartment());
    System.out.println("- Years Of Experience: " + hrEmployee1.getYearsOfExperience());
    System.out.println("- Tasks: ");
    hrEmployee1.assignTask("Interview a candidate");
    hrEmployee1.assignTask("Review employee benefits package");
    hrEmployee1.assignTask("Conduct a performance review");
    hrEmployee1.conductInterview();

    MarketingEmployee marketingEmployee1 = new MarketingEmployee("Emily", 4, "Product X");
    System.out.println("*** 4. MarketingEmployee");
    System.out.println("- Name: " + marketingEmployee1.getName());
    System.out.println("- Department: " + marketingEmployee1.getDepartment());
    System.out.println("- Product: " + marketingEmployee1.getProduct());
    System.out.println("- Tasks: ");
    marketingEmployee1.assignTask("Create a social media campaign");
    marketingEmployee1.assignTask("Design a product brochure");
    marketingEmployee1.createCampaign();
}
```

The End!