Java SE 8 Programming Language

Lab Guides

Document Code	**25e-BM/HR/HDCV/FSOFT**
Version	**1.1**
Effective Date	**20/11/2012**

Hanoi, 06/2019

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	01/Oct/2018	Create new	Create new	DieuNT1	VinhNV
2	01/Jun/2019	Update template	Fsoft template	DieuNT1	VinhNV

Contents

FRESHER ACADEMY

CODE:	**JPL.M.L101**
TYPE:	**Medium**
LOC:	
DURATION:	**60 MINUTES**

Unit 5: Basic OOP

Lab Guide 1: Abstract class, Inheritance, Polymorphism

Objectives: JPL-9

- Able to create Java-based applications that take advantage of Java object-oriented features, including encapsulation, inheritance, and polymorphism.
- Understand the static components in java

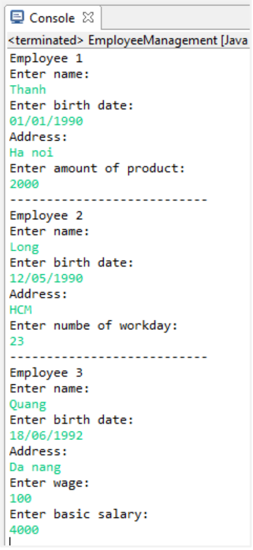
Problem Descriptions:

Create a new package named **fa.training.abstraction** in **JPL.M.L101** project.

- Create an abstract class **Employee** with: *employee name*, *date of birth*, *address, company name*, **calcSalary**() abstract method, **inputData**() method to input employee information, **display**() method to show all information. This class contains constructor and getter/setter if needs.
- There are three employee type:
  - Production staff: amount of product, salary = amount of product * 20$;
  - Daily staff: number of workdays, salary = number of workdays * 15$;
  - Manager: basic salary, wage, salary = basic salary * wage;
- Create class **ProductionStaff**, **DailyStaff** and **Manager** that extends **Employee** class. Override **calcSalary** to calculate salary for each employee type.
- Create another class named **EmployeeManagement** that creates an array contains 3 employees of above type, loop to calls the **display**() method to print the outputs.

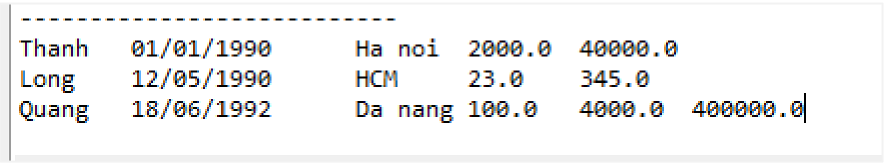
Screen Designs:

Screen 1: input data



```
<terminated> EmployeeManagement [Java
Employee 1
Enter name:
Thanh
Enter birth date:
01/01/1990
Address:
Ha noi
Enter amount of product:
2000
---------------------------
Employee 2
Enter name:
Long
Enter birth date:
12/05/1990
Address:
HCM
Enter numbe of workday:
23
---------------------------
Employee 3
Enter name:
Quang
Enter birth date:
18/06/1992
Address:
Da nang
Enter wage:
100
Enter basic salary:
4000
```

Screen 2: display data



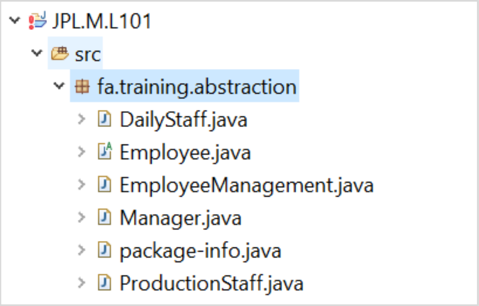
```
-----------------------------
Thanh   01/01/1990     Ha noi  2000.0   40000.0
Long    12/05/1990     HCM     23.0     345.0
Quang   18/06/1992     Da nang 100.0    4000.0   400000.0
```

<u>Functional Requirements:</u>

- ✓ Run the program and explains result.
- ✓ You change the program by adds validation methods to check input data before assign to objects. Demo with n employees.

<u>Guidelines:</u>

Project struture:



```
JPL.M.L101
  src
    fa.training.abstraction
      DailyStaff.java
      Employee.java
      EmployeeManagement.java
      Manager.java
      package-info.java
      ProductionStaff.java
```

✓ **ProductStaff** class

```java
package fa.training.abstraction;

import java.util.Scanner;

public class ProductionStaff extends Employee {

    private static final int UNIT_PRICE = 20;

    private double amountOfProduct;

    @Override
    public double calcSalary() {
        return amountOfProduct * UNIT_PRICE;
    }

    @Override
    protected void inputData(Scanner scanner) {
        /*
         * Call inputData method from parent class.
         */
        super.inputData(scanner);

        System.out.println("Enter amount of product: ");
        amountOfProduct = Double.parseDouble(scanner.nextLine());

        System.out.println("---------------------------");
    }

    @Override
    protected void display() {
        // Call method of parent class
        super.display();

        System.out.print("\t" + amountOfProduct + "\t" + this.calcSalary() + "\n");
    }

}
```

- ✓ **Employee** class

```java
package fa.training.abstraction;

import java.util.Scanner;

public abstract class Employee {
    private String employeeName;
    private String dateOfBirth;
    private String address;
    private static String companyName;

    public static String getCompanyName() {
        return companyName;
    }

    public static void setCompanyName(String companyName) {
        Employee.companyName = companyName;
    }
    /**
     * User enter data.
     *
     * @param scanner
     */
    protected void inputData(Scanner scanner) {

        System.out.println("Enter name: ");
        employeeName = scanner.nextLine();

        System.out.println("Enter birth date: ");
        dateOfBirth = scanner.nextLine();

        System.out.println("Address: ");
        address = scanner.nextLine();
    }

    /**
     * Display data to console.
     */
    protected void display() {
        System.out.print(employeeName + "\t" + dateOfBirth + "\t" + address +
                                                                      "\t" + companyName);
    }
    // getter/setter method

    public String getEmployeeName() {
        return employeeName;
    }

    public void setEmployeeName(String employeeName) {
        this.employeeName = employeeName;
    }

    public String getDateOfBirth() {
        return dateOfBirth;
    }

    public void setDateOfBirth(String dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public abstract double calcSalary();

}
```

✓ **DailyStaff** class

```java
package fa.training.abstraction;

import java.util.Scanner;

public class DailyStaff extends Employee {
    private static final int WAGE_DAY = 15;

    private double numbeOfWorkday;

    @Override
    public double calcSalary() {
        return numbeOfWorkday * WAGE_DAY;
    }

    @Override
    protected void inputData(Scanner scanner) {
        /*
         * Call inputData method from parent class.
         */
        super.inputData(scanner);

        System.out.println("Enter number of workday: ");
        numbeOfWorkday = Double.parseDouble(scanner.nextLine());

        System.out.println("---------------------------");
    }

    @Override
    protected void display() {
        // Call method of parent class
        super.display();

        System.out.print("\t" + numbeOfWorkday + "\t" + this.calcSalary() + "\n");
    }

}
```

✓ **Manager** class

```java
package fa.training.abstraction;

import java.util.Scanner;

public class Manager extends Employee {
    private double wage;
    private double basicSalary;

    @Override
    public double calcSalary() {
        return wage * basicSalary;
    }

    @Override
    protected void inputData(Scanner scanner) {
        /*
         * Call inputData method from parent class.
         */
        super.inputData(scanner);

        System.out.println("Enter wage: ");
        wage = Double.parseDouble(scanner.nextLine());

        System.out.println("Enter basic salary: ");
        basicSalary = Double.parseDouble(scanner.nextLine());

        System.out.println("---------------------------");
    }

    @Override
    protected void display() {
        // Call method of parent class
        super.display();

        System.out.print("\t" + wage + "\t" + basicSalary + "\t" +
                                                   this.calcSalary() + "\n");
    }
}
```

✓ **EmployeeManagement** class

```java
package fa.training.abstraction;

import java.util.Scanner;

public class EmployeeManagement {

    public static void main(String[] args) {

        Employee employees[] = new Employee[3];

        // Create 3 objects
        ProductionStaff productionStaff = new ProductionStaff();
        DailyStaff dailyStaff = new DailyStaff();
        Manager manager = new Manager();

        Scanner scanner = new Scanner(System.in);

        // Call inputData method
        System.out.println("Employee 1");
        productionStaff.inputData(scanner);

        System.out.println("Employee 2");
        dailyStaff.inputData(scanner);

        System.out.println("Employee 3");
        manager.inputData(scanner);

        Employee.setCompanyName("FPT");

        // Push to existed array
        employees[0] = productionStaff;
        employees[1] = dailyStaff;
        employees[2] = manager;

        // loop
        for (Employee employee : employees) {
            // An instance of Polymorphims
            employee.display();
        }

        scanner.close();
    }
}
```

-- THE END -