

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №6
По дисциплине «СПП» за 5 семестр

Выполнил:

Студент группы ПО-3

Ковалёва А. И.

Проверил:

Крощенко А. А.

Вариант 12

Цель: приобрести навыки применения паттернов проектирования при решении практических задач с использованием языка Java.

Задание 1:

Заводы по производству автомобилей. Реализовать возможность создавать автомобили различных типов на различных заводах.

Паттерн: Абстрактная фабрика.

Текст программы:

```
package com.company;

public class Main {
    public static void main(String[] args) {
        CarFactory audiFactory = new AudiFactory();
        CarFactory bmwFactory = new bmwFactory();
        audiFactory.createCar().ride();
        bmwFactory.createCar().ride();
    }
}

interface Car {
    void ride();
}

interface CarFactory {
    Car createCar();
}

class Audi implements Car {
    @Override
    public void ride() {
        System.out.println("Audi is riding");
    }
}

class AudiFactory implements CarFactory {
    @Override
    public Car createCar() {
        return new Audi();
    }
}

class bmw implements Car {
    @Override
    public void ride() {
        System.out.println("Bmw is riding");
    }
}

class bmwFactory implements CarFactory {
    @Override
```

```
        public Car createCar() {  
            return new bmw();  
        }  
    }  
}
```

Результат выполнения:

```
Audi is riding  
Bmw is riding
```

Задание 2:

Проект «Универсальная электронная карта». В проекте должна быть реализована универсальная электронная карта, в которой есть функции паспорта, страхового полиса, банковской карты и т. д.

Паттерн: Декоратор.

Текст программы:

```
package com.company;  
  
interface UniversalCardInterface {  
    void functionality();  
}  
  
class UniversalCard implements UniversalCardInterface {  
    @Override  
    public void functionality() {  
        System.out.println("Universal card functionality: ");  
    }  
}  
  
class Decorator implements UniversalCardInterface {  
    private UniversalCardInterface card;  
  
    Decorator(UniversalCardInterface card) {  
        this.card = card;  
    }  
  
    @Override  
    public void functionality() {  
        card.functionality();  
    }  
}  
  
class CreditCardDecorator extends Decorator {  
    CreditCardDecorator(UniversalCardInterface card) {  
        super(card);  
    }  
  
    @Override  
    public void functionality() {  
        super.functionality();  
        System.out.println("- credit card actions");  
    }  
}
```

```

class PassportCreditDecorator extends Decorator {
    PassportCreditDecorator(UniversalCardInterface card) {
        super(card);
    }

    @Override
    public void functionality() {
        super.functionality();
        passportAction();
    }

    void passportAction() {
        System.out.println("- passport");
    }
}

class InsurancePolicyDecorator extends Decorator {
    InsurancePolicyDecorator(UniversalCardInterface card) {
        super(card);
    }
    private boolean hasPolicy = false;

    @Override
    public void functionality() {
        super.functionality();
        getInsurancePolicyAction();
    }

    void getInsurancePolicyAction() {
        System.out.println("- insurance policy");
    }
}

public class Main {
    public static void main(String[] args) {
        UniversalCardInterface card = new InsurancePolicyDecorator(new
        PassportCreditDecorator(new CreditCardDecorator(new UniversalCard())));
        card.functionality();
    }
}

```

Результат выполнения:

Universal card functionality:

- credit card actions
- passport
- insurance policy

Задание 3:

Проект «Принтеры». В проекте должны быть реализованы разные модели принтеров, которые выполняют разные виды печати.

Паттерн: стратегия

Текст программы:

```
package com.company;

public class Main {
    public static void main(String[] args) {
        Printer printer = new Printer(new LaserPrinter());
        printer.print();
        printer.turnOn();
        printer.print();
        printer.changePrinterTo(new InkPrinter());
        printer.print();
        printer.turnOff();
    }
}

class Printer {
    private PrinterInterface printer;
    private boolean isOn;

    Printer(PrinterInterface printer) {
        this.printer = printer;
    }

    void print() {
        if (isOn) {
            printer.print();
        } else {
            System.out.println("Turn on printer!");
        }
    }

    void changePrinterTo(PrinterInterface printer) {
        this.printer = printer;
    }

    void turnOff() {
        System.out.println("printer is turned off");
        isOn = false;
    }

    void turnOn() {
        System.out.println("printer is turned on");
        isOn = true;
    }
}

interface PrinterInterface {
    void print();
}

class InkPrinter implements PrinterInterface {
    @Override
    public void print() {
        System.out.println("Printing... (InkPrinter)");
    }
}
```

```
}  
  
class LaserPrinter implements PrinterInterface {  
    @Override  
    public void print() {  
        System.out.println("Printing... (LaserPrinter)");  
    }  
}
```

Результат выполнения:

```
Turn on printer!  
printer is turned on  
Printing... (LaserPrinter)  
Printing... (InkPrinter)  
printer is turned off
```

Вывод: В ходе выполненной работы приобрела навыки применения паттернов проектирования при решении практических задач с использованием языка Java.