МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИИТ

ОТЧЁТ
по лабораторной работе №6

Выполнил:
студент 3 курса
группы ПО-9
Оводок В.В.
Проверил:
Крощенко А.А.

Брест 2024

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

Вариант 7

Задание 1

Преподаватель. Класс должен обеспечивать одновременное взаимодействие с несколькими объектами класса Студент. Основные функции преподавателя – ПроверитьЛабораторнуюРаботу, ПровестиКонсультацию, ПринятьЭкзамен, ВыставитьОтметку, ПровестиЛекцию.

В паттерне "Наблюдатель" преподаватель выступает в роли "субъекта", а студенты - в роли "наблюдателей". Когда преподаватель выполняет определенное действие, такое как проведение лекции или проверка лабораторной работы, он будет уведомлять всех наблюдателей (студентов) об этом событии.

Выходные данные:

```
Вадим пришел на лекцию
Леша пришел на лекцию
Провожу лекцию
Лабораторные работы Вадим
Проверяю лабораторную работу номер 1
Проверяю лабораторную работу номер 2
Проверяю лабораторную работу номер 3
Проверяю лабораторную работу номер 4
Проверяю лабораторную работу номер 5
Проверяю лабораторную работу номер 6
Лабораторные работы Леша
Проверяю лабораторную работу номер 1
Проверяю лабораторную работу номер 2
Проверяю лабораторную работу номер 3
Вадим пришел на консультацию
Леша пришел на консультацию
Провожу консультацию
Провожу экзамен
Вадим пришел на экзамен
Студент Вадим сдал экзамен
Леша пришел на экзамен
Студент Леша не сдал экзамен
```

Краткий текст программы:

```
public class Main{
    public static void main(String[] args) {
        Student student1 = new Student("Вадим", 6);
        Student student2 = new Student("Леша", 3);
        Teacher teacher = new Teacher();
teacher.addStudent(student1);        teacher.addStudent(student2);
teacher.conductLection();        teacher.checkLabWorks(student1);
```

```
teacher.checkLabWorks(student2);
teacher.conductConsultation();
      teacher.conductExam();
   }
}
```

Задание 2

ДУ автомобиля. Реализовать иерархию автомобилей для конкретных производителей и иерархию средств дистанционного управления. Автомобили должны иметь присущие им атрибуты, и функции. ДУ имеет три основные функции – удаленная активация сигнализации, удаленное открытие/закрытие дверей и удаленный запуск двигателя. Эти функции должны отличаться по своим характеристикам для различных устройств ДУ.

Паттерн проектирования "Мост" .

Выходные данные:

```
Открытие дверей с расстояния 10м
Запуск автомобиля Mersedes
Запуск двигателя с расстояния 10м
Сигнализация активирована со звуком виу-виу-виу
Остановка автомобиля Mersedes
Остановка двигателя с расстояния 10м
Открытие дверей с расстояния 10м
-----------------------------------
Открытие дверей с расстояния 50м
Запуск автомобиля audi
Запуск двигателя с расстояния 50м
Сигнализация активирована со звуком воу-воу-воу
Остановка автомобиля audi
Запуск двигателя с расстояния 10м
Открытие дверей с расстояния 50м
```

Краткий текст программы:

```
public static void main(String[] args) {
      Car audiCar = new AudiCar(new RemoteControl());
      Car mersedesCar = new MersedesCar(new RemoteControlProMax());

      audiCar.start();
      audiCar.stop();

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

      mersedesCar.start();
      mersedesCar.stop();

   }
```

Задание 3

Проект «Пиццерия». Реализовать формирование заказ(а)ов, их отмену, а также повторный заказ с теми же самыми позициями.

Паттерн «Команда» позволяет инкапсулировать запросы в отдельные объекты, что позволяет параметризовать клиентов с разными запросами, а также поддерживать отмену операций. Этот паттерн позволяет создать команды для формирования заказов, отмены заказов и повторного заказа. Выходные данные:

```
Формирование заказа
Id -
1
Сколько пицц хотите?
2
Margarita
4 cheese
Формирование заказа
Id -
2
Сколько пицц хотите?
1
Tropicana
Отмена заказа номер 1
Повторный заказ
ID - 1
Pizza list:
Margarita
4 cheese
is cancelled
```

Краткий текст программы:

```
public class Main {
    public static void main(String[] args) {
        Order order = new Order();
        Order order1 = new Order();
        OrderHistory orderHistory = new OrderHistory();

        CreateOrderCommand createOrderCommand = new CreateOrderCommand(order,
orderHistory);
        RepeatOrderCommand repeatOrderCommand = new RepeatOrderCommand(order,
orderHistory);
        CancelOrderCommand cancelOrderCommand = new CancelOrderCommand(order);

        CreateOrderCommand createOrder1Command = new CreateOrderCommand(order1,
orderHistory);
        RepeatOrderCommand repeatOrder1Command = new RepeatOrderCommand(order1,
orderHistory);
        CancelOrderCommand cancelOrder1Command = new CancelOrderCommand(order1);

        Waiter waiter = new Waiter();
```

```
        waiter.setCommand(createOrderCommand);
waiter.pressButton();

        waiter.setCommand(createOrder1Command);
waiter.pressButton();

        waiter.setCommand(cancelOrderCommand);
waiter.pressButton();

        waiter.setCommand(repeatOrderCommand);
waiter.pressButton();

    }
}
```

Вывод: приобрел навыки применения паттернов проектирования при решении практических задач с использованием языка Java