

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ

ОТЧЁТ  
по лабораторной работе №6

Выполнил:

студент группы ПО-8  
Вейгандт И.О.

Проверил:  
Крощенко А.А.

Брест 2024

## Вариант 9(номер зачётной книжки 210629)

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

### Задание 1:

Проект «Бургер-закусочная». Реализовать возможность формирования заказа из определенных позиций (тип бургера (веганский, куриный и т.д.)), напиток (холодный – пепси, кока-кола и т.д.; горячий – кофе, чай и т.д.), тип упаковки – с собой, на месте. Должна формироваться итоговая стоимость заказа.

Для данной задачи мы можем использовать паттерн команды (Command Pattern). Этот паттерн позволяет инкапсулировать запросы (методы) в объекты, параметризовать и планировать их выполнение.

```
import java.util.Scanner;
import java.util.HashMap;
import java.util.Map;

interface Command {
    double getPrice();
    void execute();
}

class BurgerCommand implements Command {
    private final String burgerType;
    private final double price;

    public BurgerCommand(String burgerType, double price) {
        this.burgerType = burgerType;
        this.price = price;
    }

    @Override
    public double getPrice() {
        return price;
    }

    @Override
    public void execute() {
        System.out.println("Заказан бургер: " +
burgerType);
    }
}
```

```

}

class DrinkCommand implements Command {
    private final String drinkType;
    private final double price;

    public DrinkCommand(String drinkType, double price) {
        this.drinkType = drinkType;
        this.price = price;
    }

    @Override
    public double getPrice() {
        return price;
    }

    @Override
    public void execute() {
        System.out.println("Заказан напиток: " +
drinkType);
    }
}

```

```

class OrderManager {
    private final Map<String, Command> commands = new
HashMap<>();
    private double totalPrice = 0;

    public void addCommand(String itemName, Command
command) {
        commands.put(itemName, command);
    }

    public void executeCommand(String itemName) {
        Command command = commands.get(itemName);
        if (command != null) {
            command.execute();
            totalPrice += command.getPrice();
        } else {
            System.out.println("Такого элемента меню
нет.");
        }
    }

    public double getTotalPrice() {

```

```

        return totalPrice;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        OrderManager orderManager = new OrderManager();
        orderManager.addCommand("Веганский бургер", new
BurgerCommand("Веганский бургер", 5.99));
        orderManager.addCommand("Куриный бургер", new
BurgerCommand("Куриный бургер", 4.49));
        orderManager.addCommand("Пепси", new
DrinkCommand("Пепси", 1.99));
        orderManager.addCommand("Кока-кола", new
DrinkCommand("Кока-кола", 1.99));

        System.out.println("Добро пожаловать в Бургер-
закусочную!");
        System.out.println("Введите название элемента меню
(например, 'Веганский бургер' или 'Пепси'):");

        System.out.println("Выберите тип упаковки ('с
собой' или 'на месте'):");
        String packagingType = scanner.nextLine();

        while (true) {
            String itemName = scanner.nextLine();
            if (itemName.equalsIgnoreCase("Готово")) {
                break;
            }
            orderManager.executeCommand(itemName);
        }

        double totalPrice = orderManager.getTotalPrice();
        System.out.println("Итоговая стоимость заказа: " +
totalPrice + " руб.");

        System.out.println("Спасибо за заказ! Приятного
аппетита!");
    }
}

```

```

C:\Users\veiga\jdk\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\
Добро пожаловать в Бургер-закусочную!
Введите название элемента меню (например, 'Веганский бургер' или 'Пепси'):
Выберите тип упаковки ('с собой' или 'на месте'):
с собой

Такого элемента меню нет.
Пепси
Заказан напиток: Пепси
Куриный бургер
Заказан бургер: Куриный бургер
Кока-кол
Такого элемента меню нет.
Кока-кола
Заказан напиток: Кока-кола
Готово
Итоговая стоимость заказа: 8.47 руб.
Спасибо за заказ! Приятного аппетита!

```

## Задание 2:

Проект «Часы». В проекте должен быть реализован класс, который дает возможность пользоваться часами со стрелками так же, как и цифровыми часами. В классе «Часы со стрелками» хранятся повороты стрелок.

Для реализации проекта "Часы" можно применить паттерн проектирования "Наблюдатель" (Observer pattern).

```

import java.util.ArrayList;
import java.util.List;

interface Observer {
    void update(int hours, int minutes, int seconds);
}

class Clock {
    private List<Observer> observers = new ArrayList<>();
    private int hours;
    private int minutes;
    private int seconds;

    public void addObserver(Observer observer) {
        observers.add(observer);
    }
}

```

```

        public void removeObserver(Observer observer) {
            observers.remove(observer);
        }

        public void setTime(int hours, int minutes, int
seconds) {
            this.hours = hours;
            this.minutes = minutes;
            this.seconds = seconds;
            notifyObservers();
        }

        private void notifyObservers() {
            for (Observer observer : observers) {
                observer.update(hours, minutes, seconds);
            }
        }
    }

class DigitalClock implements Observer {
    @Override
    public void update(int hours, int minutes, int seconds)
    {
        System.out.printf("Цифровые часы:
%02d:%02d:%02d\n", hours, minutes, seconds);
    }
}

class AnalogClock implements Observer {
    @Override
    public void update(int hours, int minutes, int seconds)
    {
        System.out.println("Стрелочные часы: " + hours + "
часы, " + minutes + " минуты, " + seconds + " секунды");
    }
}

public class Main {
    public static void main(String[] args) {
        Clock clock = new Clock();
        DigitalClock digitalClock = new DigitalClock();
        AnalogClock analogClock = new AnalogClock();

        clock.addObserver(digitalClock);
        clock.addObserver(analogClock);
    }
}

```

```
        clock.setTime(12, 30, 0);
    }
}
```

```
C:\Users\veiga\.jdk\openjdk-19.0.1\bin\java.exe "
Цифровые часы: 12:30:00
Стрелочные часы: 12 часы, 30 минуты, 0 секунды
```

### Задание 3:

Шифрование текстового файла. Реализовать класс-шифровщик текстового файла с поддержкой различных алгоритмов шифрования. Возможные варианты шифрования: удаление всех гласных букв из текста, изменение букв текста на буквы, получаемые фиксированным сдвигом из алфавита (например, шифром буквы а будет являться буква д для сдвига 4 и т.д.), применение операции исключающее или с заданным ключом.

Для реализации класса-шифровщика текстового файла с поддержкой различных алгоритмов шифрования, можно воспользоваться паттерном проектирования "Стратегия" (Strategy pattern).

```
import java.io.*;
import java.util.Scanner;

interface EncryptionStrategy {
    String encrypt(String text);
}

class RemoveVowelsEncryption implements EncryptionStrategy
{
    @Override
    public String encrypt(String text) {
        return text.replaceAll("[aeiouAEIOU]", "");
    }
}

class CaesarCipherEncryption implements EncryptionStrategy
{
    private final int shift;

    public CaesarCipherEncryption(int shift) {
        this.shift = shift;
    }

    @Override
    public String encrypt(String text) {
```

```

        StringBuilder encryptedText = new StringBuilder();
        for (char c : text.toCharArray()) {
            if (Character.isLetter(c)) {
                char base = Character.isLowerCase(c) ? 'a'
: 'A';
                encryptedText.append((char) ((c - base +
shift) % 26 + base));
            } else {
                encryptedText.append(c);
            }
        }
        return encryptedText.toString();
    }
}

```

```

class XOREncryption implements EncryptionStrategy {
    private final String key;

    public XOREncryption(String key) {
        this.key = key;
    }

    @Override
    public String encrypt(String text) {
        StringBuilder encryptedText = new StringBuilder();
        for (int i = 0; i < text.length(); i++) {
            encryptedText.append((char) (text.charAt(i) ^
key.charAt(i % key.length())));
        }
        return encryptedText.toString();
    }
}

```

```

class TextFileEncryptor {
    private EncryptionStrategy strategy;

    public TextFileEncryptor(EncryptionStrategy strategy) {
        this.strategy = strategy;
    }

    public void encryptFile(File inputFile, File
outputFile) {
        try (Scanner scanner = new Scanner(inputFile);
            PrintWriter writer = new
PrintWriter(outputFile)) {
            while (scanner.hasNextLine()) {

```



```

        String line = scanner.nextLine();
        String encryptedLine =
strategy.encrypt(line);
        writer.println(encryptedLine);
    }
    System.out.println("Файл успешно
закодирован.");
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}

public class Main {
    public static void main(String[] args) {
        TextFileEncryptor encryptor;

        EncryptionStrategy strategy = new
RemoveVowelsEncryption();
        // EncryptionStrategy strategy = new
CaesarCipherEncryption(3);
        // EncryptionStrategy strategy = new
XOREncryption("key");

        encryptor = new TextFileEncryptor(strategy);
        File inputFile = new File("input.txt");
        File outputFile = new File("output.txt");

        encryptor.encryptFile(inputFile, outputFile);
    }
}


```

input.txt – Блокнот

Файл Правка Формат Вид Справка

Computers have become an integral part of our daily lives.  
 They have a great impact on the way we live, work, and communicate.  
 Computers have opened up new possibilities.  
 Due to the Internet, students have access to information beyond traditional textbooks.  
 They can conduct research, collaborate with peers on projects, expanding their knowledge horizons.  
 In today's world, being computer literate is essential for future success.  
 By integrating computers into education, students can learn how to navigate digital tools,  
 analyze and evaluate online information, and develop problem-solving and coding skills.

C:\Users\veiga\.jdk\openjdk-19.0.1\bin\java.exe  
 Файл успешно закодирован.

 output.txt – Блокнот

Файл Правка Формат Вид Справка

---

```
Cmptrs hv bcm n ntgrl prt f r dly lvs.  
Thy hv grt mpct n th wy w lv, wrk, nd cmmnct.  
Cmptrs hv pnd p nw pssblts.  
D t th ntrnt, stdnts hv ccss t nfrmtn bynd trdtnl txtbks.  
Thy cn cndct rsrch, cllbrt wth prs n prjcts, xpndng thr knwldg hrzns.  
n tdy's wrld, bng cmptr ltrt s ssntl fr ftr scss.  
By ntgrtng cmptrs nt dctn, stdnts cn lrn hw t nvgt dgtl tls,  
nlyz nd vlt nln nfrmtn, nd dvlp prblm-slvng nd cdng skills.
```

**Вывод:** приобрёл навыки применения паттернов проектирования при решении практических задач с использованием языка Java.