

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №6

Специальность ПО-9

Выполнил
А. С. Мисюк,
студент группы ПО-9
Проверил
А. А. Крощенко,
ст. преп. кафедры ИИТ,
«__» _____ 2024 г.

Брест 2024

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

Вариант №4 (210664)

Для заданий был выбран паттерн *Компоновщик*.

Задание 1. Проект «Туристическое бюро». Реализовать возможность выбора программы тура (проезд, проживание, питание, посещение музеев, выставок, экскурсии и т.д.). Должна формироваться итоговая стоимость заказа.

Код программы:

```
package com.labrab6.task1;

//...
public class Main
{
    public static void main(String[] args)
    {
        // Создаем компоненты
        TourComponent transportation = new Transportation(100.0);
        TourComponent accommodation = new Accommodation(200.0);
        TourComponent meals = new Meals(50.0);
        TourComponent museumVisits = new MuseumVisits(30.0);
        TourComponent excursions = new Excursions(80.0);

        // Создаем группу компонентов и добавляем в нее компоненты
        TourPackage tourPackage = new TourPackage();
        tourPackage.addComponent(transportation);
        tourPackage.addComponent(accommodation);
        tourPackage.addComponent(meals);
        tourPackage.addComponent(museumVisits);
        tourPackage.addComponent(excursions);

        // Рассчитываем итоговую стоимость заказа
        double totalCost = tourPackage.calculateCost();
        System.out.println("Итоговая стоимость заказа: " + totalCost);
    }
}

package com.labrab6.task1;

/**
 * Общий интерфейс компонент
 */
interface TourComponent
{
    double calculateCost();
}

package com.labrab6.task1;

class TourPackage implements TourComponent {
    private List<TourComponent> components = new ArrayList<>();

    public void addComponent(TourComponent component) {
        components.add(component);
    }

    public void removeComponent(TourComponent component) {
        components.remove(component);
    }
}
```

```

    }

    @Override
    public double calculateCost() {
        double totalCost = 0.0;
        for (TourComponent component : components) {
            totalCost += component.calculateCost();
        }
        return totalCost;
    }
}

package com.labrab6.task1;

@SuppressWarnings("FieldMayBeFinal")
class Meals implements TourComponent {
    private double cost;

    public Meals(double cost) {
        this.cost = cost;
    }

    @Override
    public double calculateCost() {
        return cost;
    }
}

// и другие компоненты по подобию...

```

Пример

```

>java.exe com.labrab6.task1.Main
Итоговая стоимость заказа: 460.0

```

Задание 2. Проект «Файловая система». Реализуйте модель работы файловой системы. Должна поддерживаться иерархичность ФС на уровне директорий и отдельных файлов. Файлы могут иметь все основные присущие им атрибуты (размер, расширение, дата создания и т.д.).

Код программы:

```

package com.labrab6.task2;

/**...*/
public class Main
{
    public static void main(String[] args)
    {
        // Создаем файлы
        FileSystemComponent file1 = new File("file1.txt", 100);
        FileSystemComponent file2 = new File("file2.txt", 200);
        FileSystemComponent file3 = new File("file3.txt", 150);

        // Создаем директории
        Directory directory1 = new Directory("Directory 1");
        Directory directory2 = new Directory("Directory 2");

        // Добавляем файлы в первую директорию
        directory1.addComponent(file1);
        directory1.addComponent(file2);
    }
}

```

```

        // Добавляем файлы во вторую директорию
        directory2.addComponent(file3);

        // Добавляем вторую директорию в первую
        directory1.addComponent(directory2);

        // Выводим информацию о файловой системе
        directory1.displayInfo();
    }
}

package com.labrab6.task2;

public interface FileSystemComponent
{
    default void displayInfo() { displayInfo(""); }

    void displayInfo(String indent);
}

package com.labrab6.task2;

public class File implements FileSystemComponent
{
    protected String name;
    protected int size;

    public File(String name, int size)
    {
        this.name = name;
        this.size = size;
    }

    @Override
    public void displayInfo(String indent)
    {
        System.out.println(indent + "File: " + name + ", Size: " + size + " bytes");
    }
}

package com.labrab6.task2;

public class Directory implements FileSystemComponent
{
    protected String name;
    protected List<FileSystemComponent> components = new ArrayList<>();

    public Directory(String name)
    {
        this.name = name;
    }

    public void addComponent(FileSystemComponent component)
    {
        components.add(component);
    }

    public void removeComponent(FileSystemComponent component)
    {
        components.remove(component);
    }

    @Override
    public void displayInfo(String indent)
    {
        System.out.println(indent + "Directory: " + name);
    }
}

```

```

        for (FileSystemComponent component : components)
        {
            component.displayInfo(indent + " ");
        }
    }
}

```

Пример

```

>java.exe com.labrab6.task2.Main
Directory: Directory 1
  File: file1.txt, Size: 100 bytes
  File: file2.txt, Size: 200 bytes
Directory: Directory 2
  File: file3.txt, Size: 150 bytes

```

Задание 3. Реализовать вывод ФС из 2-й группы заданий. Вывод файлов/директорий должен осуществляться в случайном порядке. Вывести основные атрибуты каждого файла/директории.

Код программы:

```

package com.labrab6.task3;

/** ... */
public class Main
{
    public static void main(String[] args)
    {
        // Создаем файлы
        FileSystemComponentTask3 file1 = new FileTask3("file1.txt", 100);
        FileSystemComponentTask3 file2 = new FileTask3("file2.txt", 200);
        FileSystemComponentTask3 file3 = new FileTask3("file3.txt", 150);

        // Создаем директории
        DirectoryTask3 directory1 = new DirectoryTask3("Directory 1");
        DirectoryTask3 directory2 = new DirectoryTask3("Directory 2");

        // Добавляем файлы в первую директорию
        directory1.addComponent(file1);
        directory1.addComponent(file2);

        // Добавляем файлы во вторую директорию
        directory2.addComponent(file3);

        // Добавляем вторую директорию в первую
        directory1.addComponent(directory2);

        // Выводим информацию о файловой системе
        directory1.displayInfo();
    }
}

package com.labrab6.task3;

import com.labrab6.task2.FileSystemComponent;

public interface FileSystemComponentTask3 extends FileSystemComponent
{
    int getSize();
}

```

```

package com.labrab6.task3;

public class FileTask3 extends File implements FileSystemComponentTask3
{
    public FileTask3(String name, int size)
    {
        super(name, size);
    }

    @Override
    public int getSize()
    {
        return size;
    }
}

package com.labrab6.task3;

public class DirectoryTask3 extends Directory implements FileSystemComponentTask3
{
    public DirectoryTask3(String name)
    {
        super(name);
    }

    @Override
    public int getSize()
    {
        int size = 0;

        for (FileSystemComponent component : components)
        {
            if (component instanceof FileSystemComponentTask3 comp)
            {
                size += comp.getSize();
            }
        }

        return size;
    }

    @Override
    public void displayInfo(String indent)
    {
        System.out.println(indent + "Directory: " + name + ", Size: " + getSize() + " bytes");
        for (FileSystemComponent component : components)
        {
            component.displayInfo(indent + "  ");
        }
    }
}

```

Пример

```

>java.exe com.labrab6.task3.Main
Directory: Directory 1, Size: 450 bytes
  File: file1.txt, Size: 100 bytes
  File: file2.txt, Size: 200 bytes
Directory: Directory 2, Size: 150 bytes
  File: file3.txt, Size: 150 bytes

```

Рисунки с результатами работы программы

```
IDEA Community Edition 2020.1.2 (build 200.1110.0) - UTF-8 - UTF-8
C:\Users\misij\Documents\SPP\labrab6\out\production\labrab6 com.labrab6.task1.Main
Итоговая стоимость заказа: 460.0
```

```
C:\Users\misij\Documents\SPP\labrab6\out\production\labrab6 com.labrab6.task2.Main
Directory: Directory 1
File: file1.txt, Size: 100 bytes
File: file2.txt, Size: 200 bytes
Directory: Directory 2
File: file3.txt, Size: 150 bytes
```

```
C:\Users\misij\Documents\SPP\labrab6\out\production\labrab6 com.labrab6.task3.Main
Directory: Directory 1, Size: 450 bytes
File: file1.txt, Size: 100 bytes
File: file2.txt, Size: 200 bytes
Directory: Directory 2, Size: 150 bytes
File: file3.txt, Size: 150 bytes
```

Вывод: исследовал создание классов и объектно-ориентированное программирование на языке программирования Java с помощью паттернов проектирования, паттерна проектирования Компоновщик, приобрел практические навыки в области объектно-ориентированного проектирования в использовании паттернов проектирования.