

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 *Строитель*.

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

Код программы:

```
package com.labrab6.task1;

/**...
 */
public class Main
{
    public static void main(String[] args)
    {
        TourBuilder tourBuilder = new TourPackageBuilder();
        TourDirector tourDirector = new TourDirector(tourBuilder);
        Tour tour = tourDirector.createTour();
        tour.showTourDetails();
    }
}

package com.labrab6.task1;

public class Tour
{
    private String transport;
    private String accommodation;
    private String meals;
    private String activities;

    public void setTransport(String transport)
    {
        this.transport = transport;
    }

    public void setAccommodation(String accommodation)
    {
        this.accommodation = accommodation;
    }

    public void setMeals(String meals)
    {
        this.meals = meals;
    }

    public void setActivities(String activities)
    {
        this.activities = activities;
    }

    public void showTourDetails()
    {
        System.out.println("Детали тура:");
        System.out.println("Транспорт: " + transport);
    }
}
```

```

        System.out.println("Проживание: " + accommodation);
        System.out.println("Питание: " + meals);
        System.out.println("Активности: " + activities);
    }
}

```

```
package com.labrab6.task1;
```

```

public interface TourBuilder
{
    void buildTransport();

    void buildAccommodation();

    void buildMeals();

    void buildActivities();

    Tour getTour();
}

```

```
package com.labrab6.task1;
```

```

@SuppressWarnings("FieldMayBeFinal")
public class TourPackageBuilder implements TourBuilder
{
    private Tour tour;

    public TourPackageBuilder()
    {
        tour = new Tour();
    }

    @Override
    public void buildTransport()
    {
        tour.setTransport("Авиабилеты");
    }

    @Override
    public void buildAccommodation()
    {
        tour.setAccommodation("Отель 4 звезды");
    }

    @Override
    public void buildMeals()
    {
        tour.setMeals("Завтрак включен");
    }

    @Override
    public void buildActivities()
    {
        tour.setActivities("Посещение музеев, экскурсии");
    }

    @Override
    public Tour getTour()
    {
        return tour;
    }
}

```

```

package com.labrab6.task1;

@SuppressWarnings("FieldMayBeFinal")
public class TourDirector
{
    private TourBuilder tourBuilder;

    public TourDirector(TourBuilder tourBuilder)
    {
        this.tourBuilder = tourBuilder;
    }

    public Tour createTour()
    {
        tourBuilder.buildTransport();
        tourBuilder.buildAccommodation();
        tourBuilder.buildMeals();
        tourBuilder.buildActivities();
        return tourBuilder.getTour();
    }
}

```

Пример

```

>java.exe com.labrab6.task1.Main
Детали тура:
Транспорт: Авиабилеты
Проживание: Отель 4 звезды
Питание: Завтрак включен
Активности: Посещение музеев, экскурсии

```

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);
    }

    public List<FileSystemComponent> getComponents()
    {
        return components;
    }
}

```

```

    }

    @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;

import ...

import java.util.Arrays;

/**...
 */
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);

        // Создаем итератор
    }
}

```

```

RandomFileSystemComponentIterator iterator = new
RandomFileSystemComponentIterator(directory1);

    // Выводим информацию о файловой системе
    while (iterator.hasNext()) {
        FileSystemComponent component = iterator.next();
        component.displayInfo();
    }

}

}

package com.labrab6.task3;

import com.labrab6.task2.Directory;
import com.labrab6.task2.FileSystemComponent;

import ...

@SuppressWarnings("FieldMayBeFinal")
public class RandomFileSystemComponentIterator
{
    private List<FileSystemComponent> components;
    private int currentIndex;

    public RandomFileSystemComponentIterator(FileSystemComponent root)
    {
        components = new ArrayList<>();
        collectComponents(root);
        shuffleComponents();
        currentIndex = 0;
    }

    private void collectComponents(FileSystemComponent component)
    {
        components.add(component);
        if (component instanceof Directory directory)
        {
            for (FileSystemComponent subComponent : directory.getComponents())
            {
                collectComponents(subComponent);
            }
        }
    }

    private void shuffleComponents()
    {
        long seed = System.nanoTime();
        Collections.shuffle(components, new Random(seed));
    }

    public boolean hasNext()
    {
        return currentIndex < components.size();
    }

    public FileSystemComponent next()
    {
        if (hasNext())
        {
            FileSystemComponent component = components.get(currentIndex);
            currentIndex++;
            return component;
        }
        throw new IllegalStateException("No more elements");
    }
}

```

Пример

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

Рисунки с результатами работы программы

```
C:\Users\misij\Documents\SPP\labrab6\out\production\labrab6 com.labrab6.task1.Main
Детали тура:
Транспорт: Авиабилеты
Проживание: Отель 4 звезды
Питание: Завтрак включен
Активности: Посещение музеев, экскурсии
```

```
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
File: file1.txt, Size: 100 bytes
File: file3.txt, Size: 150 bytes
Directory: Directory 2
    File: file3.txt, Size: 150 bytes
Directory: Directory 1
    File: file1.txt, Size: 100 bytes
    File: file2.txt, Size: 200 bytes
Directory: Directory 2
    File: file3.txt, Size: 150 bytes
File: file2.txt, Size: 200 bytes
```

Вывод: исследовал создание классов и объектное-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.