National University of Computer and Emerging Sciences

Laboratory manual # 09
For
Software Design and Architecture

Course Instructor	Amir Iqbal
Lab Instructor	Muhammad Hashir Mohsineen, Syeda Aina Batool
Email	hashir.mohsineen@lhr.nu.edu.pk ainnie.batool275@gmail.com
Section	BSE-4D
Date	04-24-24 (MM/DD/YY)
Semester	Spring 24

Instructions for lab submission:
You have to submit source files along with a word document. In the word document you have to give the heading of each exercise/question, then paste your code. Save your word document in the following format: roll number-lab no-section i.e. 21I-0008-lab09-BSE4D.

Objective:

- ☐ Structural Design Patterns
- ☐ Composite Design Pattern
- ☐ Adapter Design Pattern

Software for this lab:

- Java netbeans
- StarUML

Create class diagrams and write Java code for the following:

Note: Write the main function for all exercises.

1. **Exercise:** **Marks: 10**

Develop a file management system for a digital asset management company. The system needs to handle various types of files, including images, videos, documents, and audio files.

Using the Composite design pattern, design a solution that allows the file management system to handle different types of files seamlessly. Consider the following details:

a. Each file type (e.g., image, video) has specific attributes and behaviors associated with it, such as data (String), duration, and file format.
b. Files can be organized into hierarchical structures, such as folders to facilitate navigation and management.
c. The system should support operations such as adding files, removing files, and printFileDetail.
d. The design should allow for extensibility, enabling the addition of new file types and functionalities in the future without disrupting existing features.

2. **Exercise:** **Marks: 10**

You're working on a project to develop a weather application that provides weather updates to users. You decided to integrate with a third-party weather API to fetch weather data. However, the API uses a different interface format compared to what your application expects.
Using the Adapter design pattern (class adapter) , design a solution to seamlessly integrate the third-party weather API with your weather application. Consider the following details:

a. The third-party weather API uses a non-standard interface format.

```
class ThirdPartyWeatherAPI {
    double fetchTemperature() { }
    double fetchHumidity() { }
    double fetchWindSpeed() { }
    double fetchPrecipitation() { }
}
```

b. Your weather application expects a unified interface for retrieving weather information, with methods such as getTemperature(), getHumidity(), getWindSpeed(), and getPrecipitation().
c. The adapter should translate calls from the standard interface of your weather application to the specific interface of the third-party API, enabling smooth integration between the two systems.

3. **Exercise:** **Marks: 10**

Develop a messaging application that integrates with various messaging services, such as SMS, and email. However, each messaging service has its own unique interface for sending and receiving messages, making it challenging to create a unified interface for the messaging application.

Using the Adapter design pattern (object adapter) in Java, design a solution to seamlessly integrate the messaging services with the messaging application, ensuring compatibility with each service's specific interface. Consider the following details:

1. The messaging application expects a unified interface for sending and receiving messages, with methods for sending messages, receiving messages i.e sendMessage, receiveMessage.

2. Each messaging service (SMS, email) has its own interface for sending and receiving messages, with methods specific to each service i.e. sendSMS, receiveSMS, sendEmail, receiveEmail.
3. The adapter should translate calls from the standard interface of the messaging application to the specific interface of each messaging service, enabling seamless integration between the application and the services.