|  |  |
| --- | --- |
| **SOFTWARE DESIGN PATTERNS**  Diploma in IT  Year 2 (2024/25) Semester 4 | Week **13** |
| **1** hour |
| **Practical 9 – Adapter & Façade** | |

**OBJECTIVES**

* Implement the Adapter and Façade design patterns

**ACTIVITY 1**

Recall Tutorial 9 Activity:

You are working on a multimedia application that can play different media formats. It has a common MediaPlayer interface that can play both audio and video files. You also have two existing classes: MP3Player (that can play .mp3 audio files) and MP4Player (that can play .mp4 video files). However, their interfaces are not compatible with the MediaPlayer interface.

The following are a set of classes for the application.

A black text on a white background

Description automatically generated A black and white rectangular sign with black text

Description automatically generatedA black rectangular sign with black text

Description automatically generatedA black and white rectangular sign with black text

Description automatically generatedA white rectangular sign with black text

Description automatically generatedA black and white rectangular sign

Description automatically generatedA black and white rectangular sign with black text

Description automatically generatedA black and white rectangular sign with black text

Description automatically generated

The existing MP3Player and MP4Player classes are as follows:

**MP3Player class:**

public class MP3Player

{

public void playMP3(string mp3FileName)

{

Console.WriteLine($"Playing MP3 File: {mp3FileName}");

}

}

**MP4Player class:**

public class MP4Player

{

public void playMP4(MP4File mP4File)

{

decompress(mP4File);

Console.WriteLine($"Playing MP4 File: {mP4File.Name}");

}

public void decompress(MP4File mP4File)

{

Console.WriteLine($"Decompressing MP4 File: {mP4File.Name}");

}

}

Implement the Adapter design pattern such that running the following program will produce the desired output.

**Program:**

MediaFile soundFile = new MP3File("song.mp3");

MediaFile movieFile = new MP4File("movie.mp4");

MP3Player mP3Player = new MP3Player();

MP4Player mP4Player = new MP4Player();

MediaPlayer mp3Adapter = new MP3PlayerAdapter(mP3Player);

MediaPlayer mp4Adapter = new MP4PlayerAdapter(mP4Player);

MediaPlayer mp;

mp = mp3Adapter;

mp.play(soundFile);

mp = mp4Adapter;

mp.play(movieFile);

**Output:**

Playing MP3 File: song.mp3

Decompressing MP4 File: movie.mp4

Playing MP4 File: movie.mp4

**ACTIVITY 2**

You are developing a shopping cart system for an e-commerce website. The website has several subsystems, such as an inventory management, a payment processing system, and an order confirmation system. The website offers a 1-click shopping option that allows the user to order an item by calling a simple method, which goes through the following process:

If there is sufficient stock of the item,

If there are sufficient funds,

Reduce the stock of the item

Confirm the order

Set the delivery details

Else

Show error message

Else

Show error message

Create a Façade with the checkout(itemName, quantity) method that implements the above process. The following classes are provided:

.

**Item class:**

public class Item

{

private string name;

private double price;

private int quantity;

// constructor, getters and setters

}

**OrderSystem class:**

public class OrderSystem

{

public void confirmOrder()

{

Console.WriteLine("Order is confirmed.");

}

public void setDelivery()

{

Console.WriteLine("Delivery details are set.");

}

}

**InventorySystem class:**

public class InventorySystem

{

private Dictionary<string, Item> inventory

= new Dictionary<string, Item>();

public void addItem(Item item) {

inventory[item.Name] = item;

}

public Item getItem(string name) {

return inventory[name];

}

public bool checkAvailability(string itemName, int quantity) {

Console.WriteLine($"Checking stock for {itemName}...");

return inventory[itemName].Quantity >= quantity;

}

public void reduceStock(string itemName, int quantity)

{

Console.WriteLine(

$"Reducing stock of {itemName} by {quantity}");

inventory[itemName].Quantity -= quantity;

}

}

**PaymentSystem class:**

public class PaymentSystem

{

private double balance = 0.0;

public void addFunds(double amount) {

Console.WriteLine($"Adding ${amount:N2} to balance");

balance += amount;

Console.WriteLine($"Current funds: ${balance:N2}");

}

public bool processPayment(double amount)

{

Console.WriteLine(

$"Processing payment of ${amount:N2}...");

if (amount <= balance)

{

balance -= amount;

Console.WriteLine("Payment successful.");

Console.WriteLine($"Funds left: ${balance:N2}");

return true;

}

else

{

Console.WriteLine(

"Payment unsuccessful: insufficient funds.");

return false;

}

}

}

Running the following program should produce the desired output.

**Program:**

InventorySystem inventorySystem = new InventorySystem();

PaymentSystem paymentSystem = new PaymentSystem();

OrderSystem orderSystem = new OrderSystem();

Item pokemonPack = new Item("Pokemon Pack", 7.00, 5);

inventorySystem.addItem(pokemonPack);

paymentSystem.addFunds(20.0);

ShoppingCartFacade scFacade =

new ShoppingCartFacade(inventorySystem, paymentSystem,

orderSystem);

Console.WriteLine();

scFacade.checkout("Pokemon Pack", 2);

Console.WriteLine();

scFacade.checkout("Pokemon Pack", 2);

Console.WriteLine();

paymentSystem.addFunds(30.0);

scFacade.checkout("Pokemon Pack", 2);

Console.WriteLine();

scFacade.checkout("Pokemon Pack", 2);

**Output:**

Adding $20.00 to balance

Current funds: $20.00

Starting checkout process...

Checking stock for Pokemon Pack...

Processing payment of $14.00...

Payment successful.

Funds left: $6.00

Reducing stock of Pokemon Pack by 2

Order is confirmed.

Delivery details are set.

Starting checkout process...

Checking stock for Pokemon Pack...

Processing payment of $14.00...

Payment unsuccessful: insufficient funds.

Adding $30.00 to balance

Current funds: $36.00

Starting checkout process...

Checking stock for Pokemon Pack...

Processing payment of $14.00...

Payment successful.

Funds left: $22.00

Reducing stock of Pokemon Pack by 2

Order is confirmed.

Delivery details are set.

Starting checkout process...

Checking stock for Pokemon Pack...

Insufficient stock.