**LAB # 06**

**Task 1: Implement Factory pattern for creating pages and popups(Modals, Alerts, etc.) in website for pages separate interface defined and for popups separate interface defined.**

**Solution :**

**Page Interface**

internal interface iPages

{

void home();

void about();

}

**Popup Interface**

internal interface iPopUps

{

void Alert();

void Modal();

}

**E-commerce Page**

internal class E\_commerce : iPages

{

public void home()

{

Console.WriteLine("Welcome to our E-commerce Home page ....!");

}

public void about()

{

Console.WriteLine("Welcome to our E-commerce about page ....!");

}

}

**Sport Page**

internal class Sport : iPages

{

public void home()

{

Console.WriteLine("Welcome to our Sport Home page ....!");

}

public void about()

{

Console.WriteLine("Welcome to our Sport about page ....!");

}

}

**Warning Popup**

internal class Warning : iPopUps

{

public void Alert()

{

Console.WriteLine("This is Warning Popup Be carefull !...");

}

public void Modal()

{

Console.WriteLine("This is Warning Modals Popup");

}

}

**Information Popup**

internal class Information :iPopUps

{

public void Alert()

{

Console.WriteLine("This is just Information popup ! Be alert ....");

}

public void Modal()

{

Console.WriteLine("This is Information Modals Popup");

}

}

**Page Factory**

internal class PageFactory

{

public static iPages getPage(string pageType)

{

if (pageType == "Sport") return new Sport();

if(pageType == "Ecommerce") return new E\_commerce();

return null;

}

}

**Popup Factory**

internal class PopupFactory

{

public static iPopUps getPopup(string popupType)

{

if (popupType == "Information") return new Information();

if (popupType == "Warning") return new Warning();

return null;

}

}

**Main Method**

static void Main(string[] args)

{

string type = "";

int choice, opt;

Console.Write("1) Page\n2) Popup\n\tSelect any Operation : ");

opt = int.Parse(Console.ReadLine());

if(opt == 1)

{

Console.Write("\n1) E-commerce Page\n2) Sport Page\n\tSelect any Page type : ");

choice = int.Parse(Console.ReadLine());

if (choice == 1) type = "Ecommerce";

if (choice == 2) type = "Sport";

iPages myPage = PageFactory.getPage(type);

myPage.home();

myPage.about();

}

else if(opt == 2)

{

Console.Write("\n1) Warning Popup\n2) Information Popup\n\tSelect any Popup type : ");

choice = int.Parse(Console.ReadLine());

if (choice == 1) type = "Warning";

if (choice == 2) type = "Information";

iPopUps myPopup = PopupFactory.getPopup(type);

myPopup.Alert();

myPopup.Modal();

}

else

{

Console.WriteLine("Wrong Press try Again");

Main(args);

}

}

Text

Description automatically generated**OUTPUT :**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Task 2: Implement factory pattern for creating Ranks/Positions of employees.**

**Solution :**

**Interface**

internal interface IFlashLight

{

string getFunctionality();

}

**Abstract Class Decorator**

internal abstract class FlashLightDecorator :IFlashLight

{

private IFlashLight \_flashLight;

public FlashLightDecorator(IFlashLight flashLight)

{

\_flashLight = flashLight;

}

public virtual string getFunctionality()

{

return \_flashLight.getFunctionality();

}

}

**Flash Light Class**

internal class FlashLight : IFlashLight

{

public string getFunctionality()

{

return "Simple Flash Light";

}

}

**Class Flash Light Effect**

internal class FlashLightEffect : FlashLightDecorator

{

public FlashLightEffect(IFlashLight flashLight) : base(flashLight)

{

}

public override string getFunctionality()

{

return base.getFunctionality() + ", with Blinking Effect .....";

}

}

**Class Ring Tone Effect**

internal class RingtoneEffect : FlashLightDecorator

{

public RingtoneEffect(IFlashLight ringtone) : base(ringtone)

{

}

public override string getFunctionality()

{

return base.getFunctionality()+"\nAdding Nokia Ringtone .....";

}

}

**Main Method**

static void Main(string[] args)

{

IFlashLight light = new FlashLight();

IFlashLight flash = new FlashLightEffect(light);

IFlashLight ring = new RingtoneEffect(flash);

Console.WriteLine(ring.getFunctionality());

}

Graphical user interface, text

Description automatically generated**OUTPUT :**