using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp7

{

abstract class Button

{

public abstract void Show();

}

class WinButton : Button

{

public override void Show()

{

Console.WriteLine("Windows Button");

}

}

class MacButton : Button

{

public override void Show()

{

Console.WriteLine("Mac Button");

}

}

abstract class Checkbox

{

public abstract void Check();

}

class WinCheckbox : Checkbox

{

public override void Check()

{

Console.WriteLine("I check as Windows");

}

}

class MacCheckbox : Checkbox

{

public override void Check()

{

Console.WriteLine("I check as Mac");

}

}

interface IGuiFactory

{

Button CreateButton();

Checkbox CreateCheckbox();

}

class WinFactory : IGuiFactory

{

public Button CreateButton()

{

return new WinButton();

}

public Checkbox CreateCheckbox()

{

return new WinCheckbox();

}

}

class MacFactory : IGuiFactory

{

public Button CreateButton()

{

return new MacButton();

}

public Checkbox CreateCheckbox()

{

return new MacCheckbox();

}

}

class Application

{

private IGuiFactory \_factory;

public Application(IGuiFactory factory)

{

\_factory = factory;

}

public void CreateUI()

{

\_factory.CreateButton().Show();

\_factory.CreateCheckbox().Check();

}

}

class Program

{

static void Main(string[] args)

{

Application application = new Application(new WinFactory());

application.CreateUI();

}

}

}