using System;

namespace RefactoringGuru.DesignPatterns.Mediator.Conceptual

{

public interface IMediator

{

void Notify(object sender, string ev);

}

class AuthenticationMediator : IMediator

{

private Button \_component1;

private Textbox \_component2;

public AuthenticationMediator(Button component1, Textbox component2)

{

this.\_component1 = component1;

this.\_component1.SetMediator(this);

this.\_component2 = component2;

this.\_component2.SetMediator(this);

}

public void Notify(object sender, string ev)

{

if (ev == "A")

{

Console.WriteLine("Mediator reacts on A and triggers folowing operations:");

this.\_component2.DoC();

}

if (ev == "D")

{

Console.WriteLine("Mediator reacts on D and triggers following operations:");

this.\_component1.DoB();

this.\_component2.DoC();

}

}

}

class BaseComponent

{

protected IMediator \_mediator;

public BaseComponent(IMediator mediator = null)

{

this.\_mediator = mediator;

}

public void SetMediator(IMediator mediator)

{

this.\_mediator = mediator;

}

}

class Button : BaseComponent

{

public void DoA()

{

Console.WriteLine("Component 1 does A.");

this.\_mediator.Notify(this, "A");

}

public void DoB()

{

Console.WriteLine("Component 1 does B.");

this.\_mediator.Notify(this, "B");

}

}

class Textbox : BaseComponent

{

public void DoC()

{

Console.WriteLine("Component 2 does C.");

this.\_mediator.Notify(this, "C");

}

public void DoD()

{

Console.WriteLine("Component 2 does D.");

this.\_mediator.Notify(this, "D");

}

}

class Program

{

static void Main(string[] args)

{

// The client code.

Button component1 = new Button();

Textbox component2 = new Textbox();

new AuthenticationMediator(component1, component2);

Console.WriteLine("Client triggets operation A.");

component1.DoA();

Console.WriteLine();

Console.WriteLine("Client triggers operation D.");

component2.DoD();

}

}

}