public abstract class Colleague

{

protected IMediator \_mediator;

public Colleague(IMediator mediator)

{

\_mediator = mediator;

}

}

public class ConcreteColleagueA : Colleague

{

public ConcreteColleagueA(IMediator mediator) : base(mediator) { }

public void Send(string msg)

{

Console.WriteLine("A send message:" + msg);

\_mediator.SendMessage(this, msg);

}

public void Receive(string msg)

{

Console.WriteLine("A receive message:" + msg);

}

}

public class ConcreteColleagueB : Colleague

{

public ConcreteColleagueB(IMediator mediator) : base(mediator) { }

public void Send(string msg)

{

Console.WriteLine("B send message:" + msg);

\_mediator.SendMessage(this, msg);

}

public void Receive(string msg)

{

Console.WriteLine("B receive message:" + msg);

}

}

public interface IMediator

{

void SendMessage(Colleague caller, string msg);

}

public class ConcreteMediator : IMediator

{

public ConcreteColleagueA Colleague1 { get; set; }

public ConcreteColleagueB Colleague2 { get; set; }

public void SendMessage(Colleague caller, string msg)

{

if (caller == Colleague1)

Colleague2.Receive(msg);

else

Colleague1.Receive(msg);

}

}