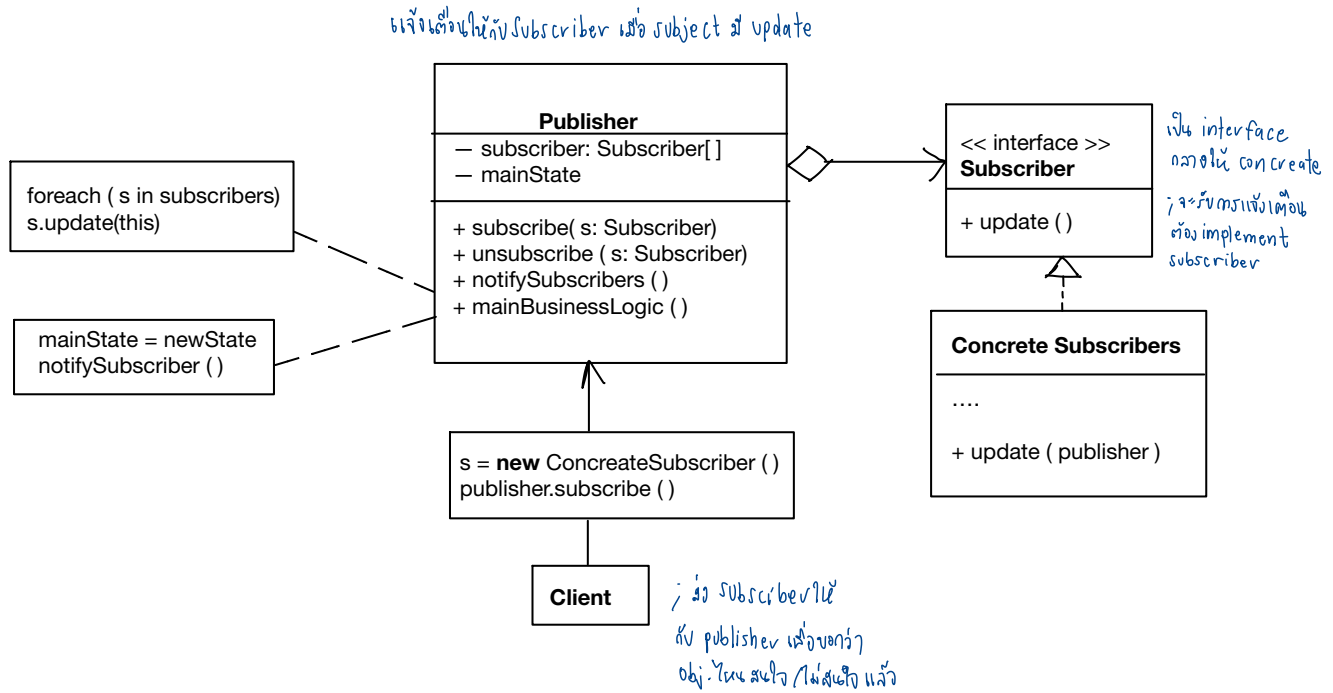


Ex. แจ้งเตือนว่ามีบ้านหรือไม่



```
foreach(var it in subscribers)
{
    it.Update(msg);
}
}

// Client
class Program
{
    static void Main(string[] args)
    {
        var student = new Student();
        var teacher = new Teacher();

        var publisher = new SchoolMessenger();
        publisher.Subscribe(student);
        publisher.Subscribe(teacher);

        Console.WriteLine("Class Annoucement: ");
        publisher.SendMessage("Today have a new homework!");

        Console.WriteLine("Student want to unsubscribe a message");
        publisher.Unsubscribe(student);

        Console.WriteLine("Class Annoucement: ");
        publisher.SendMessage("Student unsubscribed");
    }
}
```

Result Size: 541 x 576

Get your own C# server

```
Class Annoucement:
Student receive new message, Today have a new homework!
Teacher receive new message, Today have a new homework!
Student want to unsubscribe a message
Class Annoucement:
Teacher receive new message, Student unsubscribed
```

```
using System;
using System.Collections.Generic;

// Subscriber Interface
interface ISubscriber
{
    void Update(string msg);
}

class Student : ISubscriber
{
    public void Update(string msg)
        => Console.WriteLine($"Student receive new message, {msg}");
}
class Teacher : ISubscriber
{
    public void Update(string msg)
        => Console.WriteLine($"Teacher receive new message, {msg}");
}

// Publisher
class SchoolMessenger
{
    private IList<ISubscriber> subscribers;

    public SchoolMessenger()
        => subscribers = new List<ISubscriber>();

    public void Subscribe(ISubscriber newSubscriber)
        => subscribers.Add(newSubscriber);

    public void Unsubscribe(ISubscriber newSubscriber)
        => subscribers.Remove(newSubscriber);

    public void SendMessage(string msg)
    {
        foreach(var it in subscribers)
        {
            it.Update(msg);
        }
    }
}

// Client
class Program
{
    static void Main(string[] args)
    {
        var student = new Student();
        var teacher = new Teacher();

        var publisher = new SchoolMessenger();
        publisher.Subscribe(student);
        publisher.Subscribe(teacher);

        Console.WriteLine("Annoucement from university: ");
        publisher.SendMessage("Today isn't have a class");

        Console.WriteLine("Student want to unsubscribe a message");
        publisher.Unsubscribe(student);

        Console.WriteLine("Annoucement from university: ");
        publisher.SendMessage("already unsubscribed");
    }
}
```