**Структурные паттерны: Заместитель (Proxy) C#**

**Назначение:**

Изменение требований и эволюция системы могут вызвать необходимость внесения серьезных архитектурных изменений. Если на ранних этапах некая операция выполнялась на стороне клиента или же приложение состояло из одного процесса, то со временем исполнение операции может быть перенесено на сервер, а приложение разбито на несколько процессов. В результате возникает задача взаимодействия с удаленным процессом, реализация которой должна быть максимально похожей на локальное взаимодействие. Именно для таких целей предназначен паттерн «Заместитель».

**Proxy (заместитель)**— ​является суррогатом другого объекта и контролирует доступ к нему.​

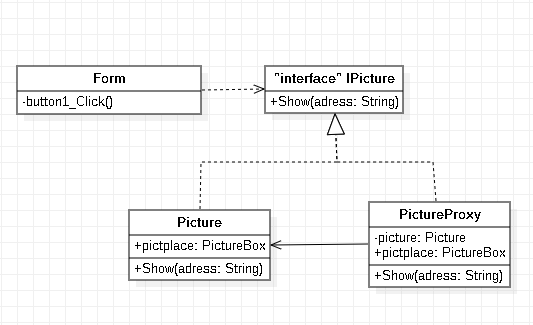
**Когда использовать Proxy?**

Существуют три основных сценария использования паттерна «Заместитель»:

* **Удаленный заместитель (remote proxies)**— ​отвечает за кодирование запроса и его аргументов для работы с компонентом в другом адресном пространстве.
* **Виртуальный заместитель (virtual proxies)**—​ может кэшировать дополнительную информацию о реальном компоненте, чтобы отложить его создание.
* **Защищающий заместитель (protection proxies)**— проверяет, имеет ли вызывающий объект необходимые для выполнения запроса права.​

**Приведённая ниже программа реализует Виртуальный заместитель (virtual proxy) для поиска различных изображений на устройстве и в интернете.**

Диаграмма классов приложения:



В данной программе при использовании объектом класса Form интерфейса IPicture происходит обращение к методу Show(adress:String) PictureProxy-объекта.  
При вызове PictureProxy-метода после завершения своих действий(в данном случае вывода на экран proxy-изображения), PictureProxy-объект обратится к методу Show(adress:String) Picture-объекта по ссылке picture: Picture. Метод Show(adress:String) Picture-объекта содержит асинхронную функцию загрузки изображения, поэтому пока изображение будет грузиться, на экране будет proxy-изображение, и только после полной загрузки появится изображение, которое нужно пользователю.

Листинг программы:

using System;

using System.Windows.Forms;

namespace ProxyApp

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

PictureProxy proxy = new PictureProxy();//создание ЗАМЕСТИТЕЛЯ

proxy.pictplace = pictureBox1;//передача заместителю ссылки на созданный на форме picturebox1

proxy.Show(textBox1.Text);//вызов метода ЗАМЕСТИТЕЛЯ

}

}

interface IPicture//интерфейс, реализуемый объектом и заместителем

{

void Show(string adress);

}

class Picture : IPicture//класс-объект

{

public PictureBox pictplace;//ссылка на picturebox для отображения картинки

public void Show(string adress)

{

pictplace.LoadAsync(adress);//асинхронная загрузка изображения по адресу с использованием собственного функционала picturebox

}

}

class PictureProxy : IPicture//класс-заместитель

{

public PictureBox pictplace;//ссылка на picturebox для отображения картинки

private Picture picture;

public void Show(string adress)

{

pictplace.Load("proxy.png");//загрузка картинки-заместителя из корневой папки проекта

if (picture == null)

{

picture = new Picture();

}

picture.pictplace = pictplace;//передача объекту ссылки на picturebox

picture.Show(adress);

}

}

}

//ссылка на gif-изображение для примера  
//https://i.gifer.com/embedded/download/OC9y.gif

**Пример работы программы:**  
На форме расположено текстовое поле. При введениии в него адрес изображения(путь к файлу на устройстве или url-адрес) и нажатии на кнопку «Открыть», приложение начнёт асинхронно подгружать соответствующее изображение, а в это время в элемент pictureBox на форме будет выведено изображение-заместитель, находящееся в корневой папке проекта.  
