



Тема: Лекция № 6

1. Слои приложения
2. Паттерн MVP
3. Dependency Injection

Что есть хорошая программа?

- Легкая поддержка
- Масштабируемость
- Повторное использование
- ...

Многослойная архитектура

Наиболее распространенный случай – **трехслойная архитектура:**

1. Слой логики представления
2. Слой бизнес-логики
3. Слой данных

Слой логики представления

- Интерфейс пользователя (UI)
- Может существовать несколько реализаций для различных платформ

Слой бизнес-логики

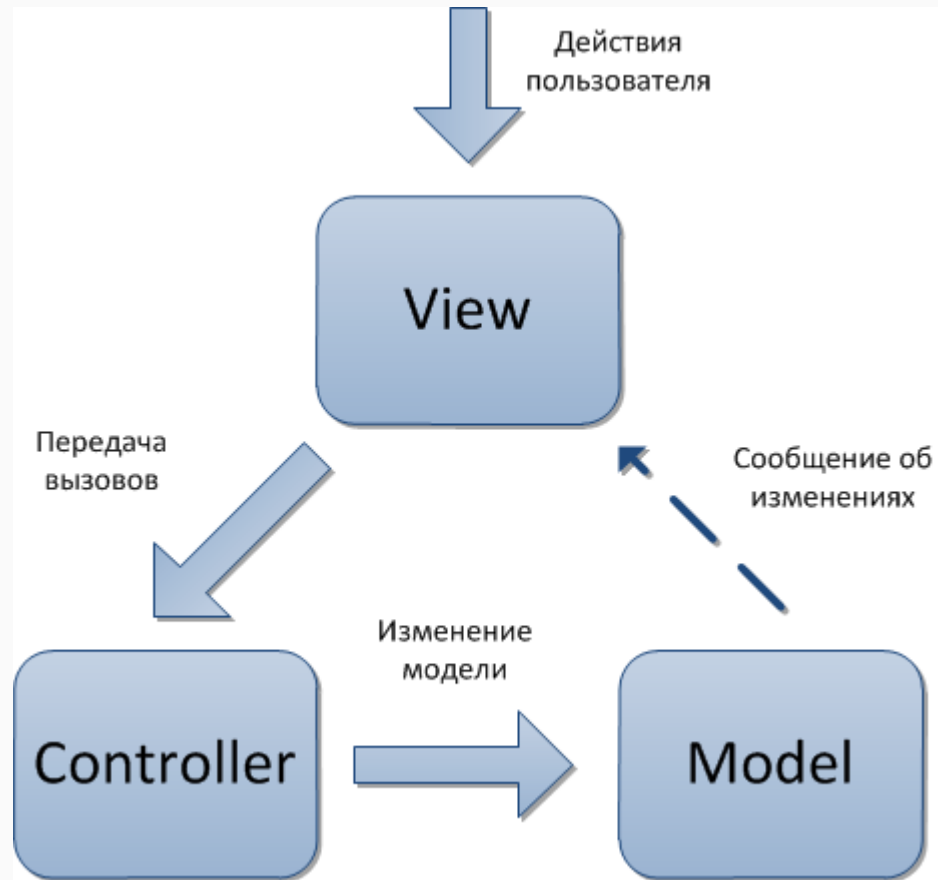
- Содержит реализацию функционала системы.
- Получает запросы от пользователя (через UI), взаимодействует со слоем данных.

Слой данных

- Хранение данных (БД)

MVC

- Model
- View
- Controller



MVP

- Model
- View
- Presenter



Пример реализации MVP в WinForms

Замена уровня представления на WPF

- Inversion of Control (инверсия управления)
- Dependency Injection (внедрение зависимостей)
- DI-контейнер

- Inversion of Control – абстрактный принцип, набор рекомендаций для написания слабо связанного кода.

- Dependency Injection – одна из реализаций IoC-принципа.

DI-контейнер

Это библиотека или фреймворк, которая упрощает и автоматизирует использование подхода инверсии контроля.

Примеры:

- Unity
- Autofac
- Ninject

Порядок работы

1) Создать экземпляр ядра Ninject

```
IKernel ninjectKernel = new StandardKernel();
```

2) Конфигурирование ядра

```
ninjectKernel.Bind<IValueCalculator>().To<ValueCalculator>();
```

3) Использование

```
IValueCalculator calc = ninjectKernel.Get<IValueCalculator>();
```

Цепочки зависимостей

```
public interface IDiscountHelper
{...}

public class DefaultDiscountHelper : IDiscountHelper
{...}

public class ValueCalculator : IValueCalculator
{
    private IDiscountHelper discountHelper;
    public ValueCalculator(IDiscountHelper discountParam)
    {
        discountHelper = discountParam;
    }
}
```


Области действия

- `InTransientScope()`
- `InSingletonScope()`
- `InThreadScope()`
- `InRequestScope()`

OnActivation/OnDeactivation

- Позволяет создавать код, который будет выполняться после создания и перед диспозом объекта.

```
Bind<IClassA>().To<ClassA>()  
.OnActivation( x => x.ActivationMethod() )  
.OnDeactivation( x =>  
x.DeactivationMethod() );
```

Спасибо за внимание!

Ссылки

- <http://outcoldman.com/ru/archive/2010/02/22/%D0%BF%D0%B0%D1%82%D1%82%D0%B5%D1%80%D0%BD%D1%8B-mvc-mvp-%D0%B8-mvvm/>
- <http://habrahabr.ru/post/131993/>
- http://professorweb.ru/my/ASP_NET/mvc/level2/2_3.php
- http://professorweb.ru/my/WPF/documents_WPF/level36/36_4.php

Install-Package Ninject -version 3.0.1.10

**Install-Package Ninject.Web.Common -
version 3.0.0.7**

**Install-Package Ninject.MVC3 -Version
3.0.0.6**