



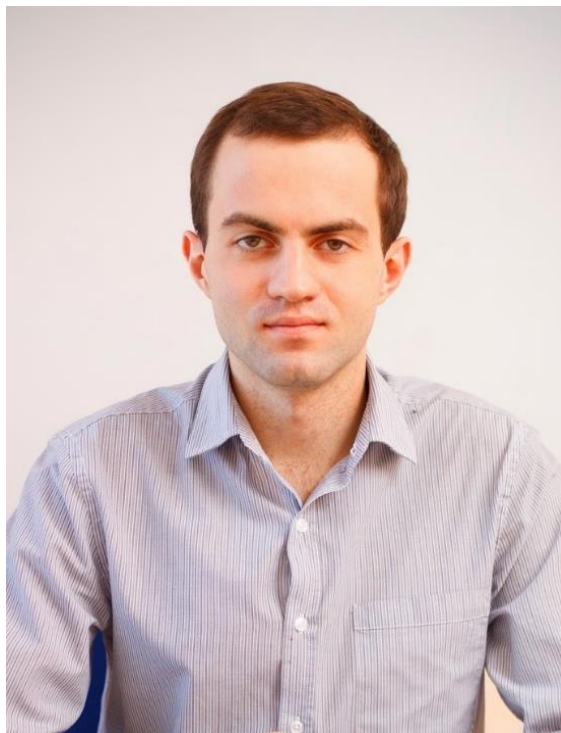
# Unit тестирование в C#

Использование Stub объектов



# Unit тестирование в C#

## Introduction



Охрименко Дмитрий  
МСТ



MCID: 9210561

# Unit тестирование в C#

Тема урока

Использование Stub объектов

# Unit тестирование в C#

## План урока

Виды unit тестов и тестовых объектов

Что такое Stub объект

Inversion of Control

# Unit тестирование в C#

## Типы тестов

**Тесты состояния** – проверяют состояние объекта после вызова метода (проверка того, что вызванный метод отработал правильно).

**Тесты взаимодействия** – проверяют правильность взаимодействия тестируемого объекта с окружением (внешними зависимостями).

**Внешняя зависимость** – это объект, с которым взаимодействует код и над которым нет прямого контроля. (Например, файловая система, потоки, память, службы и т. д.) Для ликвидации внешних зависимостей используются stubs (заглушки)

# Unit тестирование в C#

## Виды тестовых объектов

**Dummy object** – передается в тестируемый класс в качестве параметра. Не имеет поведения и с ним ничего не происходит.

**Test stub** – Используется для получения данных от внешних зависимостей, подменяя ее.

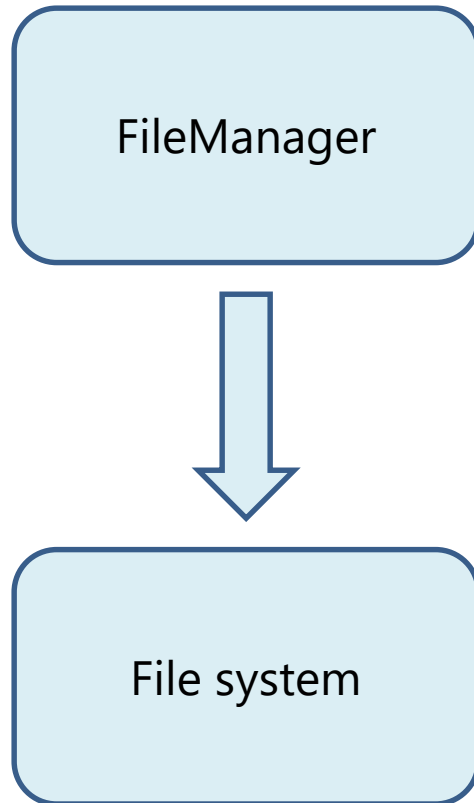
**Test spy** – используется для тестов взаимодействия. Основная задача – запись данных, которые поступают от тестируемого объекта.

**Mock object** – похож на test spy но не записывает данные от тестируемого объекта, а самостоятельно их проверяет.

**Fake object** - замена тяжеловесной внешней зависимости легковесной реализацией. Эмулятор БД в памяти, фальшивый веб сервис.

# Unit тестирование в C#

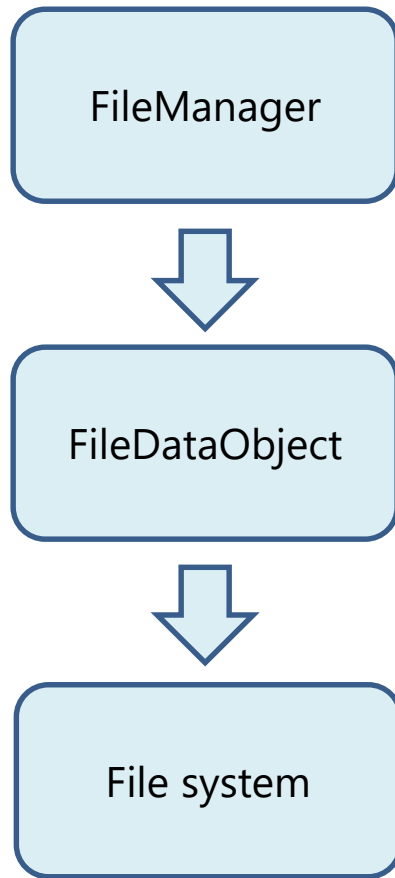
## Проблема



Тест для класса FileManager будет зависеть от файловой системы, соответственно будет являться интеграционным, так как мы будем тестировать не только FileManager, но и файловую систему (наличие файла, правильность формата файла и т.д.)

# Unit тестирование в C#

## Использование stub объекта

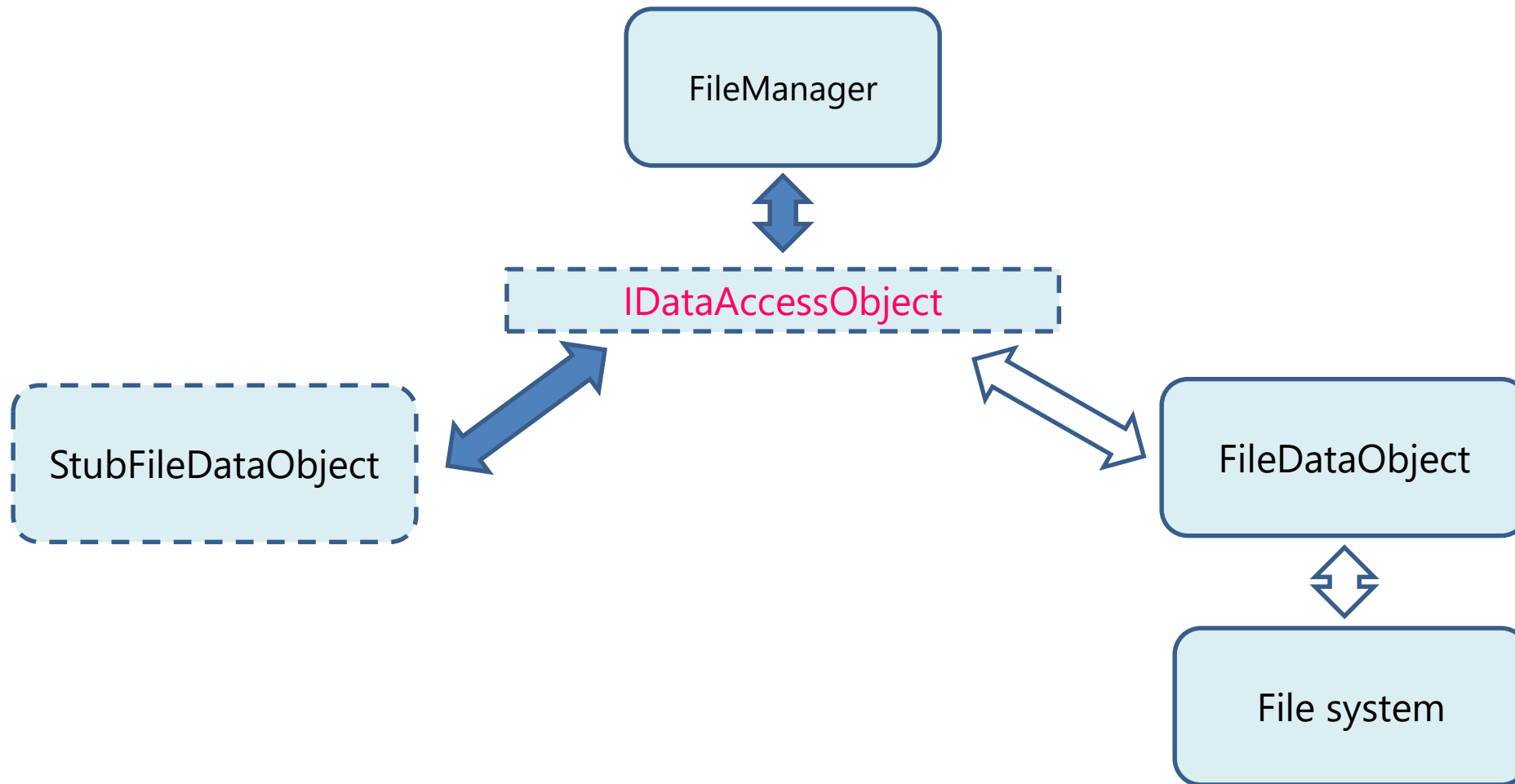


Для того, чтобы избежать прямой зависимости с файловой системой, вводится дополнительный объект (слой) `FileDataObject`, в который переносится весь код работы с файловой системой. Такая структура кода позволит заменить `FileDataObject` на подставной (stub) объект и выполнить тестирование `FileManager` без тестирования его зависимостей.



# Unit тестирование в C#

## Использование Stub объекта



# Unit тестирование в C#

## Inversion Of Control (IoC)

**Inversion of Control (инверсия управления)** — это важный абстрактный принцип, представляющий собой набор рекомендаций для написания слабо связанного кода. Суть которого заключается в том, что каждый компонент системы, должен быть как можно более изолированным от других, и не полагаться в своей работе на детали конкретной реализации других компонентов.

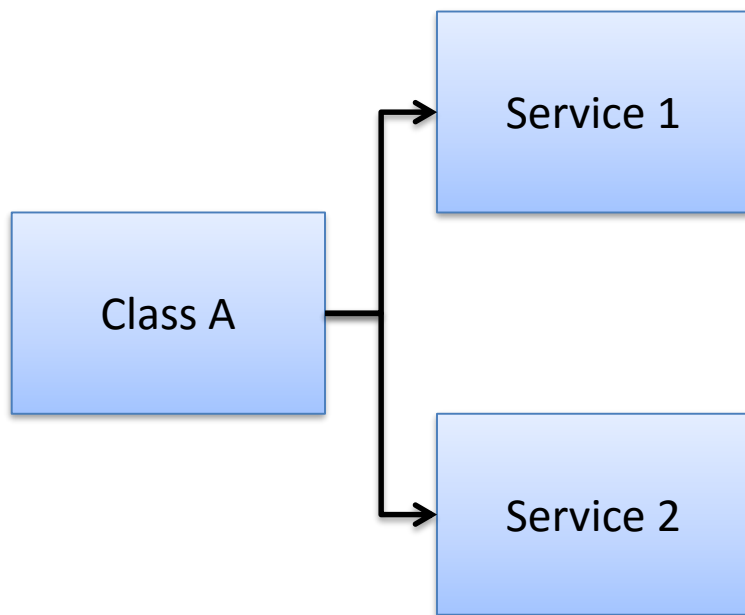
Известные реализации принципа Inversion of Control:

- Dependency Injection
- Service Locator
- Factory Method

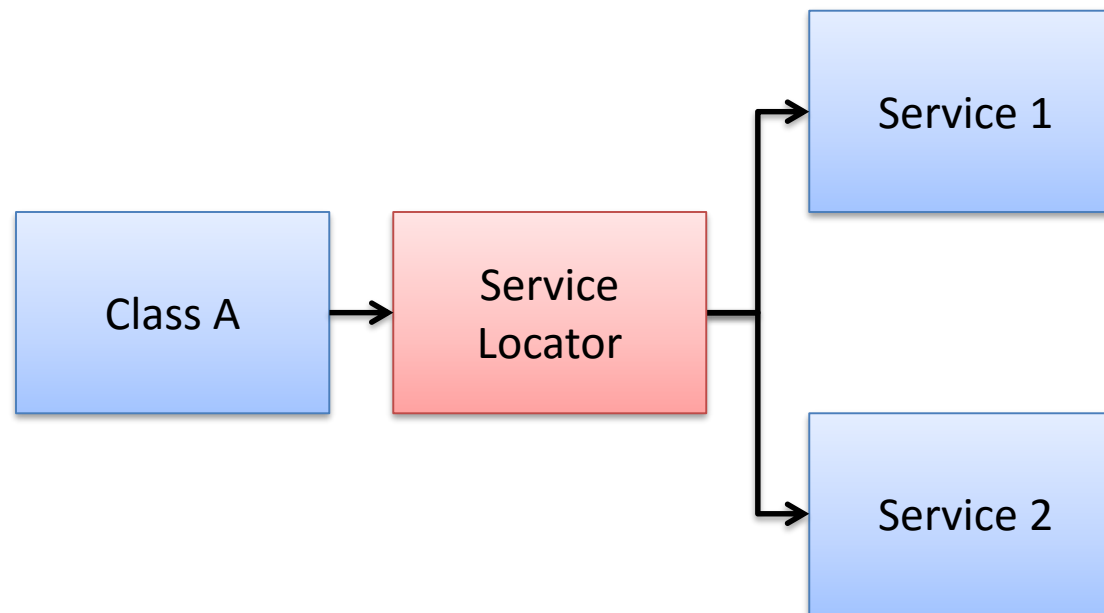
# Unit тестирование в C#

## Service Locator

**Service Locator** - паттерн использующийся для отделения объекта от его зависимостей.



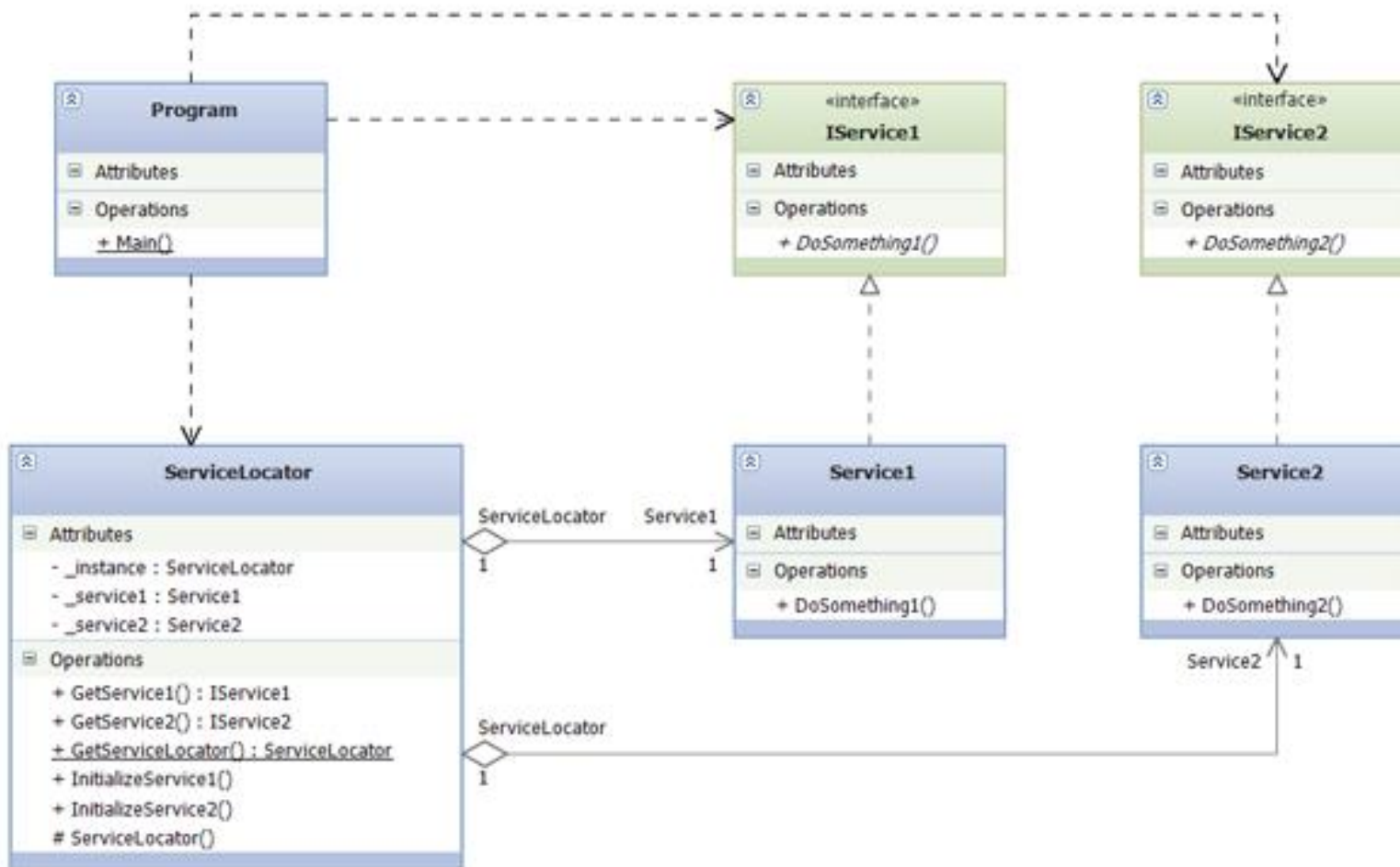
Проблема



Решение

# Unit тестирование в С#

## Service Locator – class diagram



# Unit тестирование в C#

## Dependency Injection

**Dependency Injection** – паттерн описывающий технику внедрения внешней зависимости в программный компонент.

### Способы внедрения зависимостей:

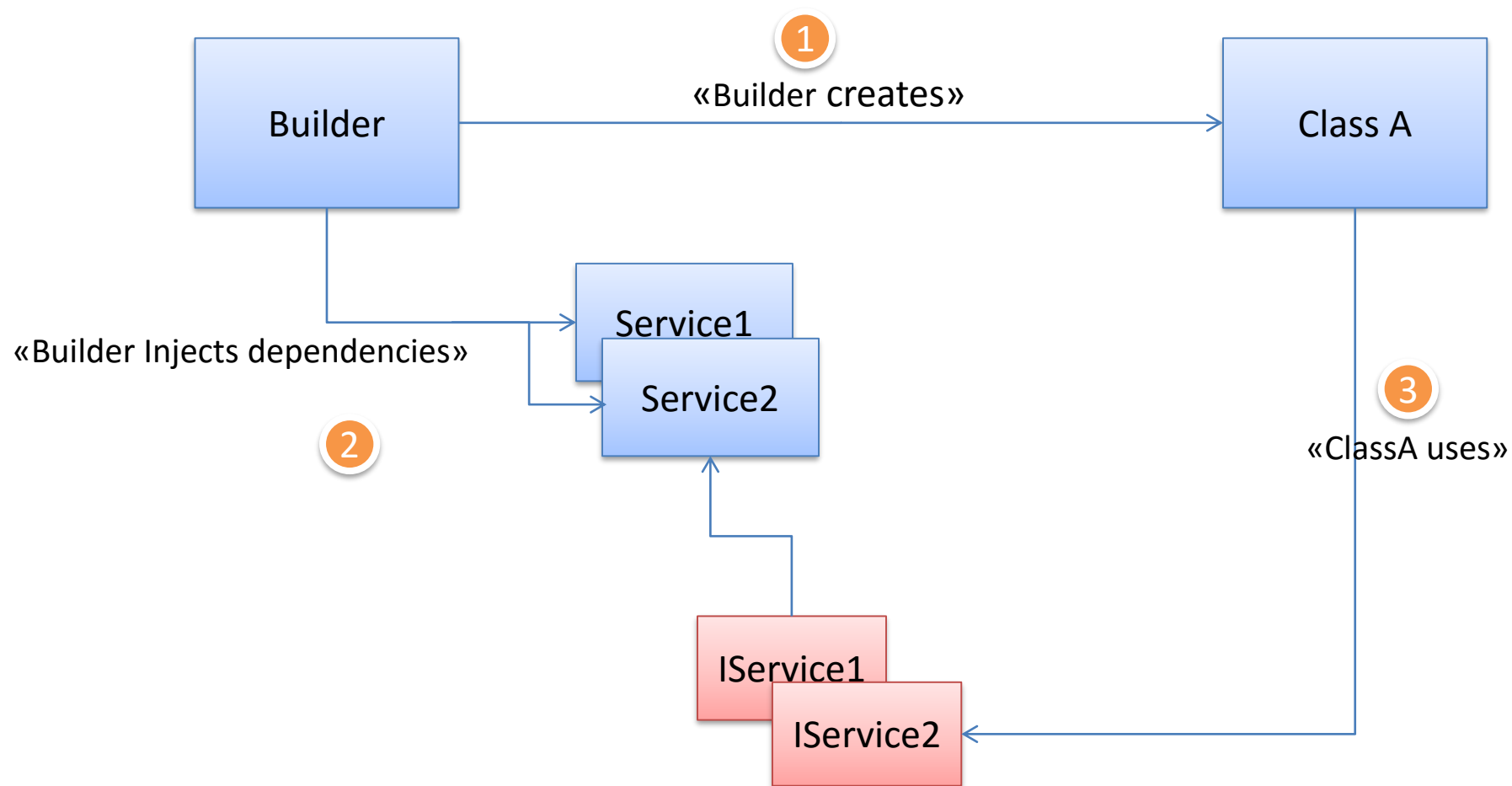
- через конструктор.
- через свойство.
- через интерфейсы.

### Способы создания экземпляров зависимостей:

Через контейнеры (Unity, Ninject, Spring .NET)

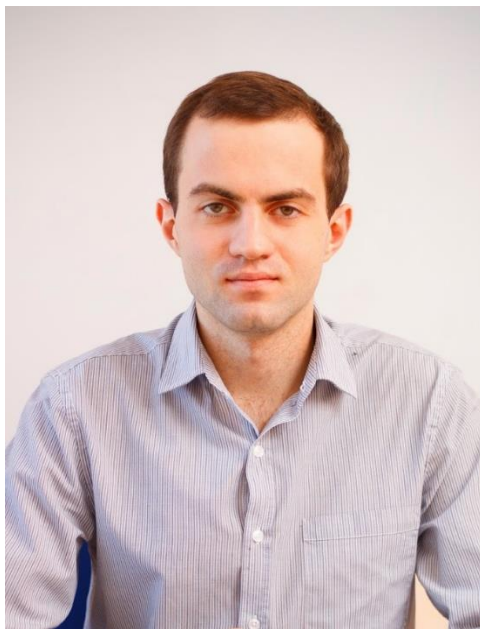
# Unit тестирование в C#

## Dependency Injection



# Unit тестирование в C#

Спасибо за внимание! До новых встреч!



Охрименко Дмитрий  
МСТ



MCID: 9210561

# Информационный видеосервис для разработчиков программного обеспечения

