

UNMANAGED .NET

СТАНИСЛАВ СИДРИСТЫЙ
ssidristyi@luxoft.com
Luxoft.com
@sidristij



LXFT
LISTED
NYSE

ЦЕЛИ ДОКЛАДА

Теоретическая часть

- Вспомнить Рихтера и red-gate добрым словом, вспомним базовые знания по памяти в .Net
- Рассмотрим модель типов .Net и поддержку наследования

Практическая часть

- Достанем скрытые структуры CLI тремя способами, без использования рефлексии
- Пробросим *object* в другой *AppDomain*
- Построим свой *ObjectPool* вне .Net памяти
- Соберем в памяти все .Net объекты

Сделать выводы

БАЗОВЫЕ ПРЕДСТАВЛЕНИЯ О ПАМЯТИ .NET



БАЗОВЫЕ ЗНАНИЯ

Области памяти:

- Есть **стек** потока – он для работы методов
- Есть **SON** – он для «маленьких» объектов (<85000B)
- Есть **LOH** – он для «больших» объектов
- Есть **Code Heap** – там размещаются результаты JIT компиляции
- Есть также **High Frequency Heap** – содержит структуры поддержки системы типов, к которым идет частое обращение (MethodTable, VTable)
- Есть еще **Low Frequency Heap** – содержит редкоиспользуемые структуры поддержки системы типов
- **Stub Heap** – содержит заглушки для COM Interop, p/invoke

НЕМНОГО ПОПОДРОБНЕЕ

Stack потока:

- Предназначен для хранения параметров методов и хранения адреса возврата из метода для каждого вызова метода
- Соответственно, при вызове метода, в стеке выделяется фрейм, в который перед вызовом складываются все параметры (включая `this` для instance методов) и адрес возврата (адрес сл. инструкции: на кот. должен перейти **ret**)

Managed Heaps:

- Размещаются все экземпляры классов и `value types` в качестве их полей
- В зависимости от размера уходят либо в SOH, либо в LOH
- Управляются механизмом Garbage Collector, который работает по-разному для SOH/LOH

И ЕЩЕ НЕМНОГО ПОПОДРОБНЕЕ

Card Table:

- Битовая карта, каждый бит которой отражает измененное состояние поля класса, которое попало в диапазон байт, за который этот бит отвечает
- Диапазон байт равен 4K
- При проходе по полям, GC проходит в поле если Gen поля ниже, либо выше, но содержит установленные биты

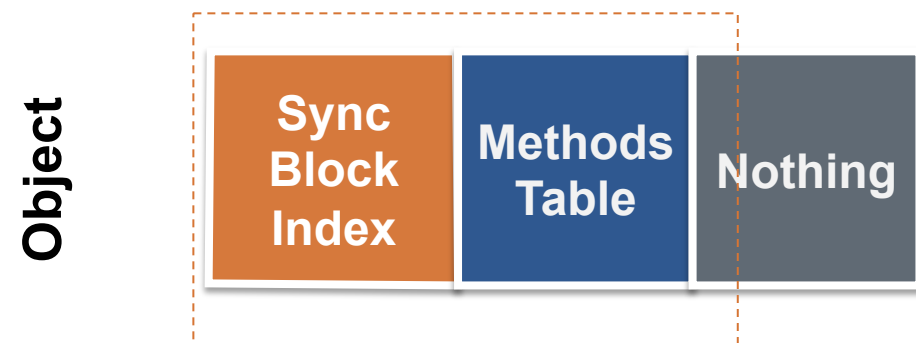
Ephemeral Segments:

- Сегменты – это группы **страниц памяти**, внутри которых размещаются кучи
- SOH куча, имеющая несколько сегментов, размещает Gen0 и Gen1 в последнем выделенном сегменте, а все остальные сегменты занимает Gen2

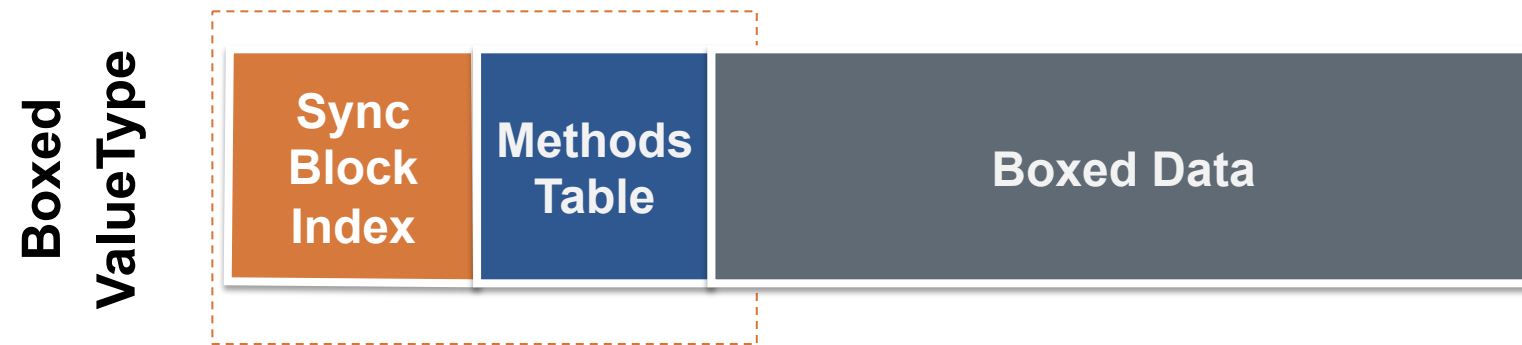
ПАМЯТЬ ПРОЦЕССОВ - ВИРТУАЛЬНАЯ

- Это значит что доступное адресное пространство – не RAM
- Часть – на диске, часть – в RAM
- Доступность не линейная, а «островками» с настройками прав доступа
- Нет понятия AppDomain – значит нет ограничения доступа между ними
- А при наличии прав, можно снять память чужого процесса

Структура типов данных



Обязательная часть



Обязательная часть



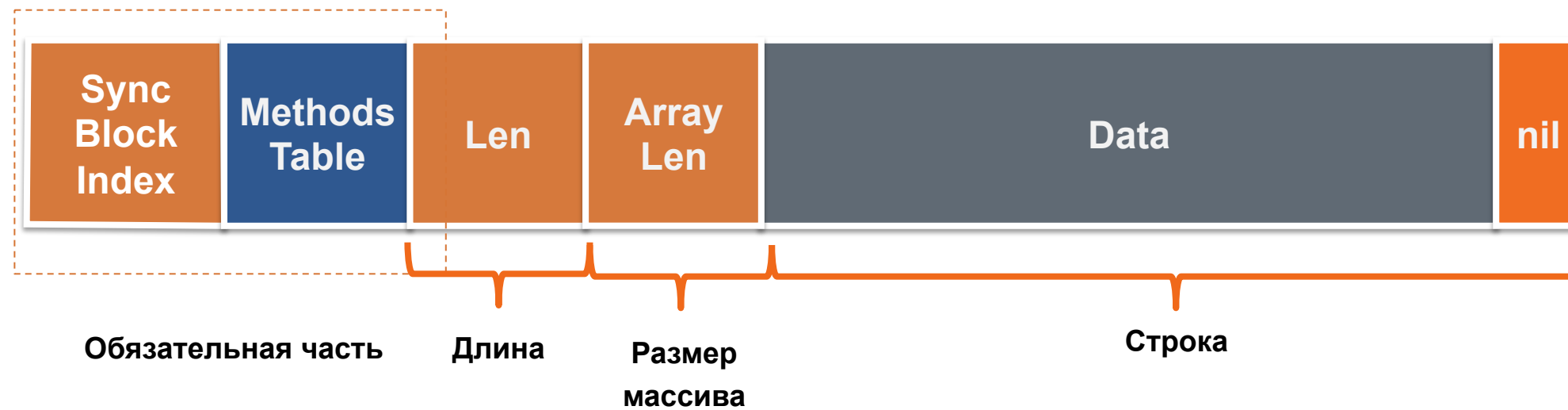
Обязательная часть

Структура типов данных

String 4.0+



String 3.5-

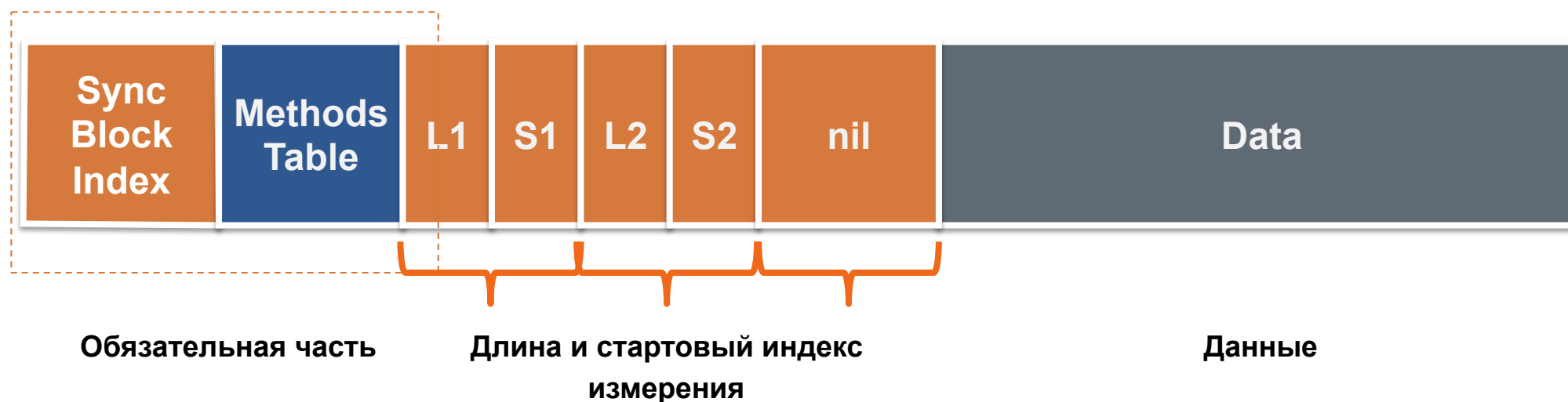


Структура типов данных

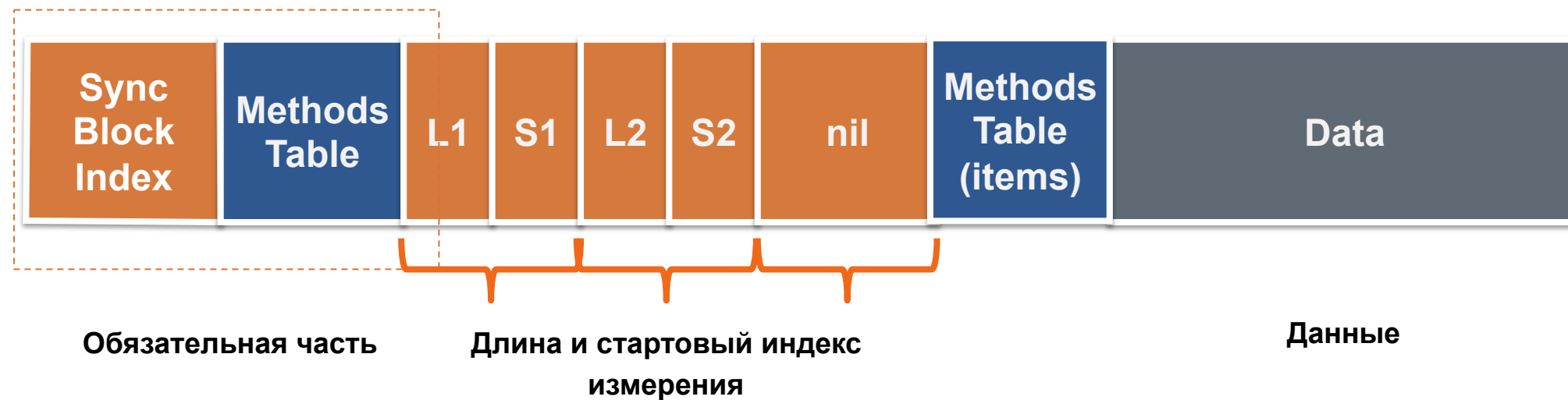
Array₁ of
ValueTypes



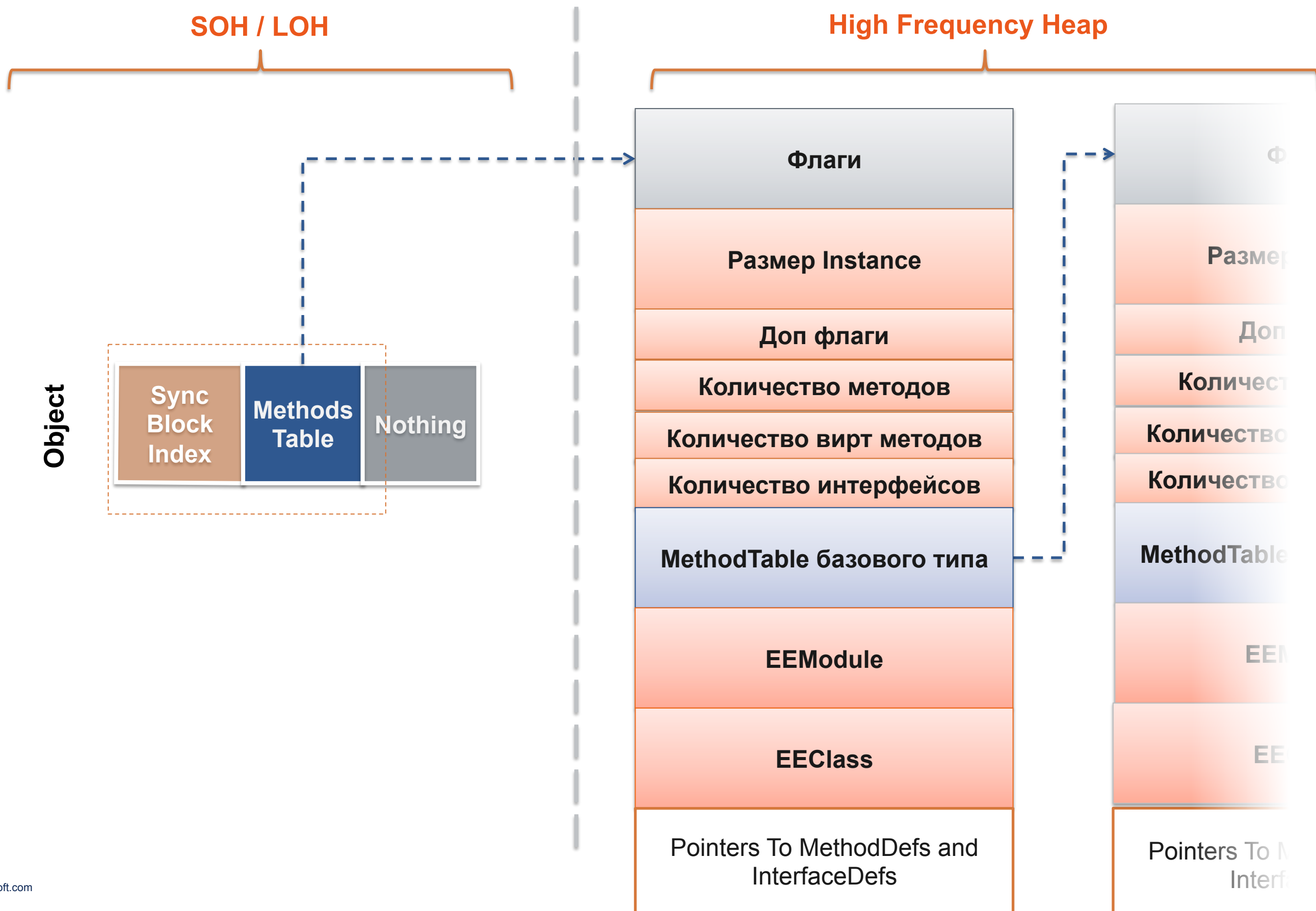
Array_N of
ValueTypes



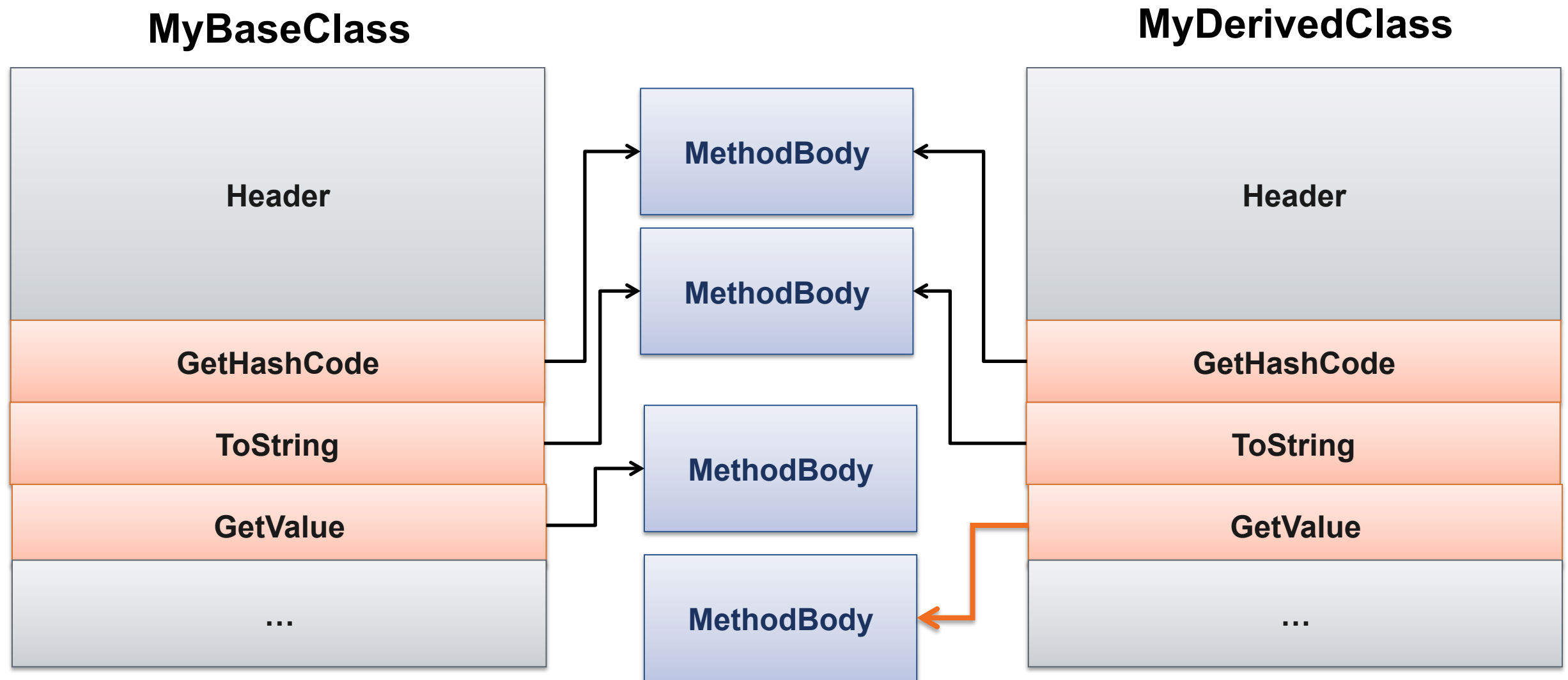
Array_N of
RefTypes



Структура типов данных



Структура типов данных

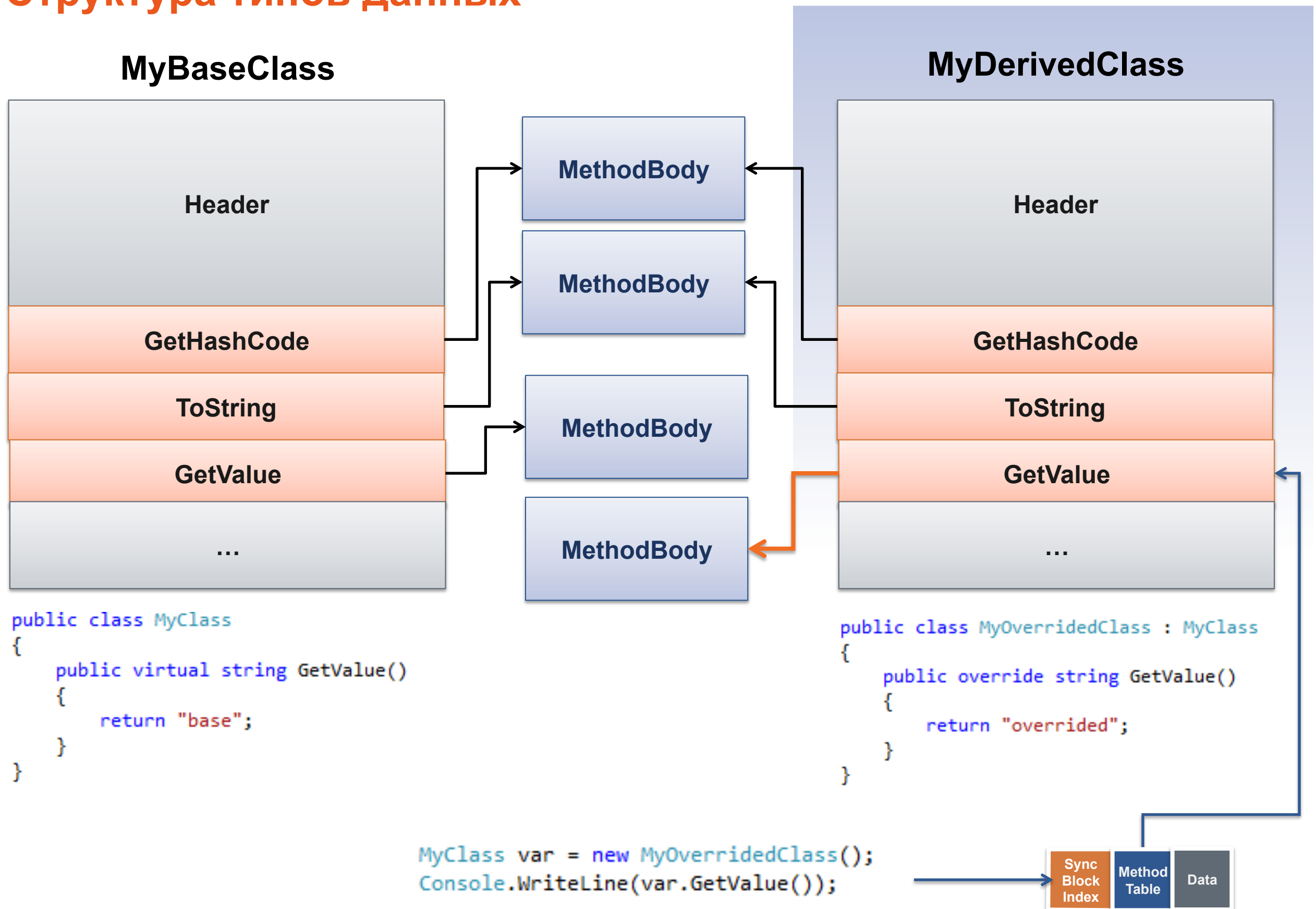


```
public class MyClass
{
    public virtual string GetValue()
    {
        return "base";
    }
}
```

```
public class MyOverriddenClass : MyClass
{
    public override string GetValue()
    {
        return "overridden";
    }
}
```

```
MyClass var = new MyOverriddenClass();
Console.WriteLine(var.GetValue());
```

Структура типов данных



ПРИМЕРЫ КОДА



ДЛЯ САМОРАЗВИТИЯ



Мастер-класс от .Net-гуру, Дино Эспозито



**«Свежий взгляд на проектирование
реального ПО: от DDD и CQRS до Event
Sourcing»**

25 октября, Москва

Подробнее на www.luxoft-training.ru

ВОПРОСЫ



Fb: @luxoft

Twitter: @sidristij, @luxoft

Luxoft.com

