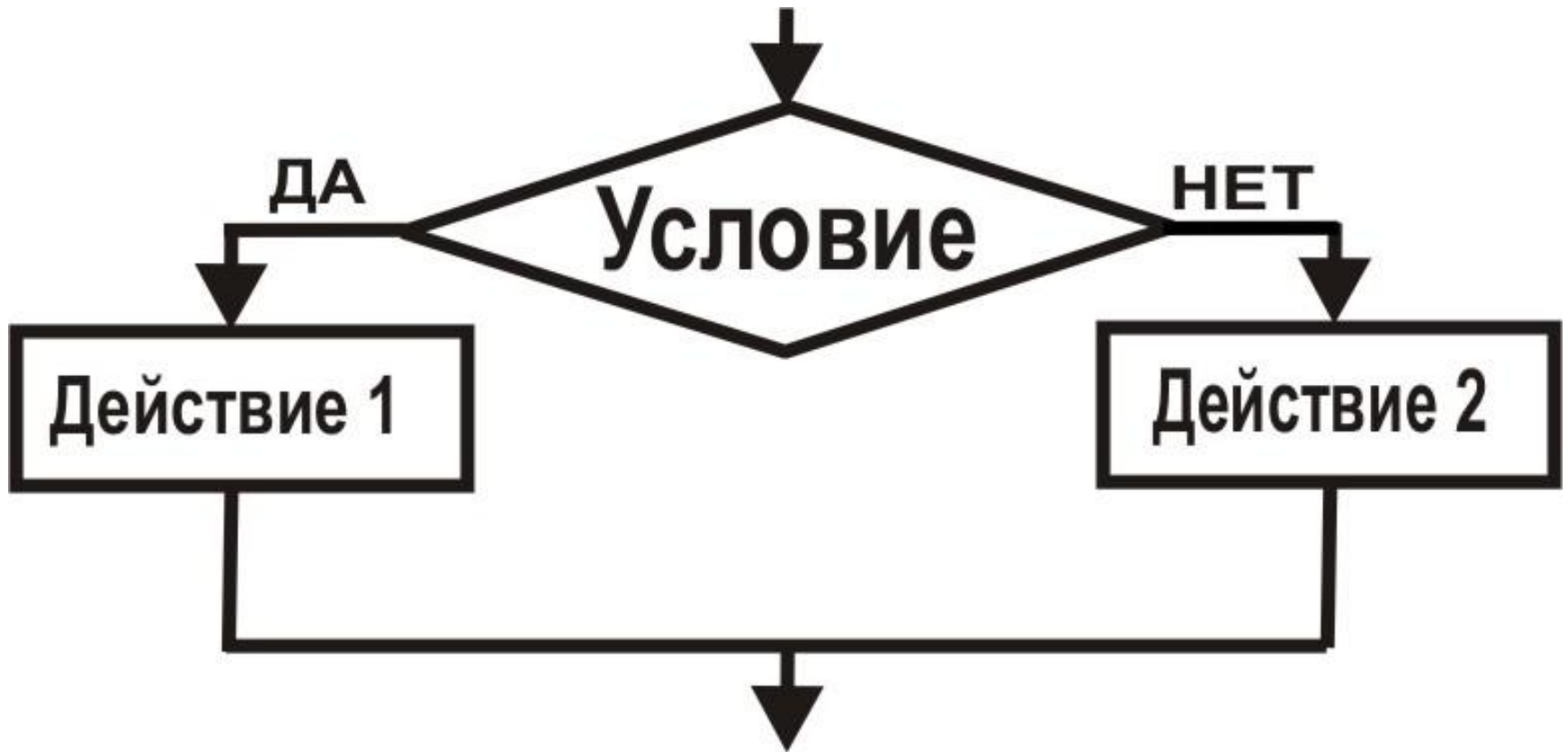


# Школа «Молодого программиста»

Погружение в C#: основы  
(базовые понятия, пространства  
имен, классы, ООП).

Что было в начале? А в начале  
был алгоритм



# Типы переменных в C#

Тип	Размер	Область значений
bool	1 байт	true или false
byte	8-битный	0 до 255
char	16-битный	Любой символ Unicode в диапазоне 0-65535
decimal	128-битный	$\pm 1,0 \times 10^{-28}$ до $\pm 7,9 \times 10^{28}$
double	64-битный	от $\pm 5,0 \times 10^{-324}$ до $\pm 1,7 \times 10^{308}$
float	32-битный	от $\pm 1,5 \times 10^{-45}$ до $\pm 3,4 \times 10^{38}$
int	32-битный	-2147483648 до 2147483647
long	64-битный	-9223372036854775808 до 9223372036854775807
sbyte	8-битный	-128 до 127
short	16-битный	-32768 до 32767
uint	32-битный	0 до 4294967295
ulong	64-битный	0 до 18446744073709551615
ushort	16-битный	0 до 65535
string	-----	текстовая переменная

# Операторы в C#

- Операторы ветвления: if-else, switch
- Циклические операторы: while, do-while, for, foreach
- Операторы сравнения: <, >, ==, ...
- Комментарии: `/**/,//`

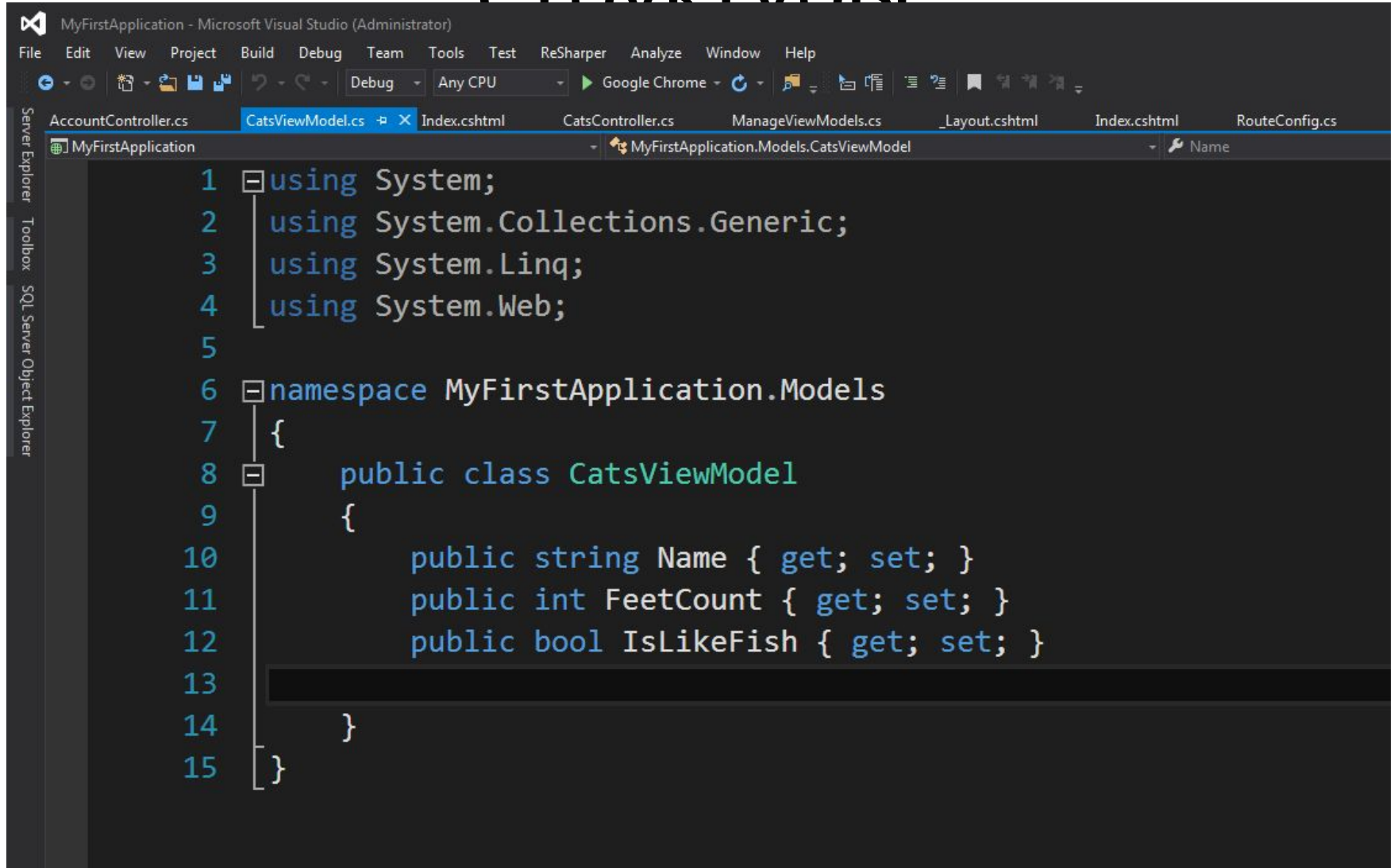
Потом была функция

anonymous function / callback method

**function()**

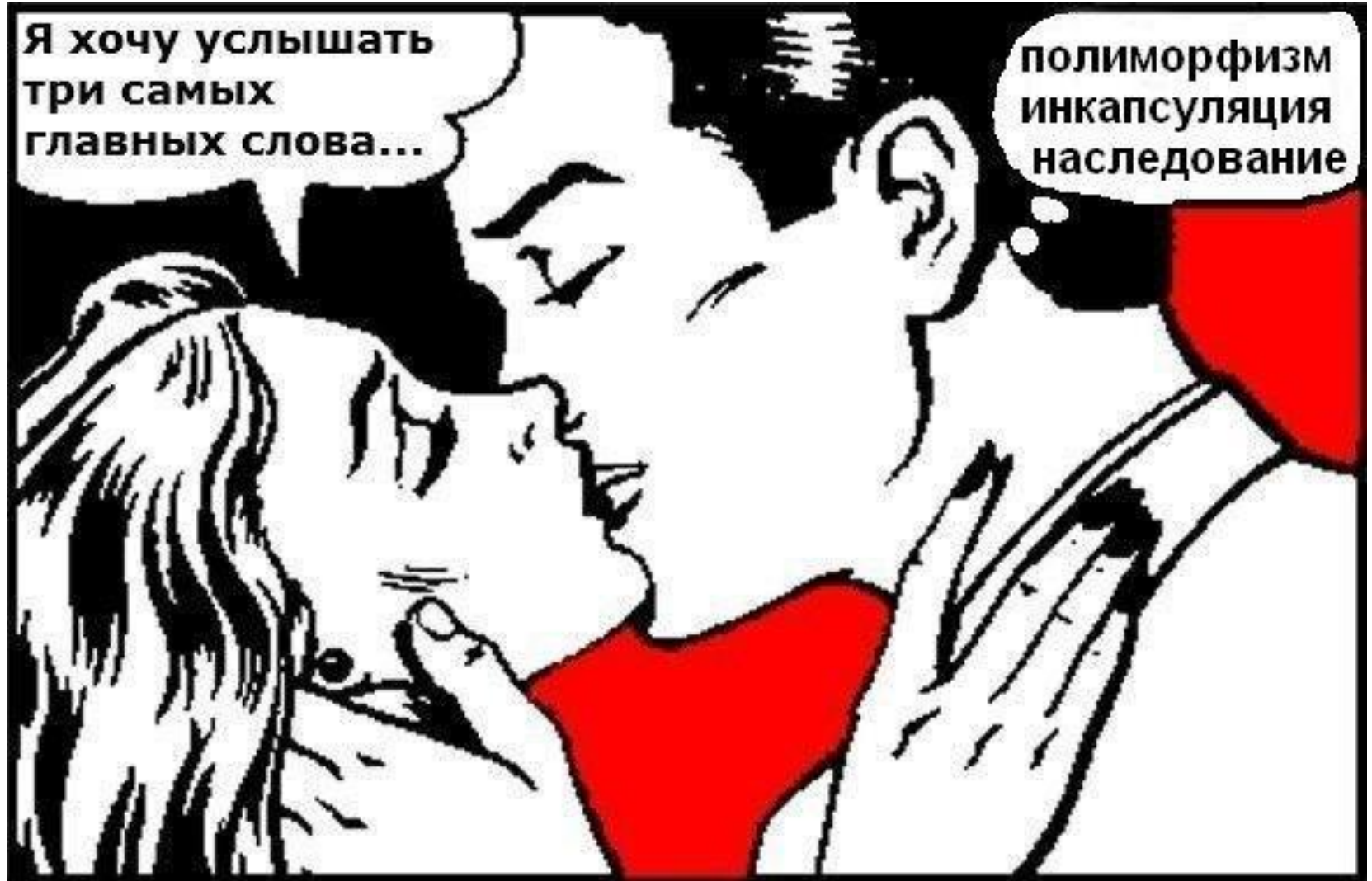
**{ }**

# Потом были классы и СТРУКТУРЫ



```
1 using System;
2     using System.Collections.Generic;
3     using System.Linq;
4     using System.Web;
5
6 namespace MyFirstApplication.Models
7 {
8     public class CatsViewModel
9     {
10         public string Name { get; set; }
11         public int FeetCount { get; set; }
12         public bool IsLikeFish { get; set; }
13
14     }
15 }
```

# Что такое ООП?



В центре ООП находится понятие  
объекта.

**Объект** — это сущность, экземпляр  
класса, которой можно посылать  
сообщения, и которая может на них  
реагировать, используя свои данные.



# Инкапсуляция как механизм контроля доступа

- **Public**
- **Protected**
- **Private**

# Наследование

**КОНКРЕТНЫЙ ПАЦАН**



**АБСТРАКТНЫЙ ПАЦАН**



# Полиморфизм

Прямой руль



Райзер

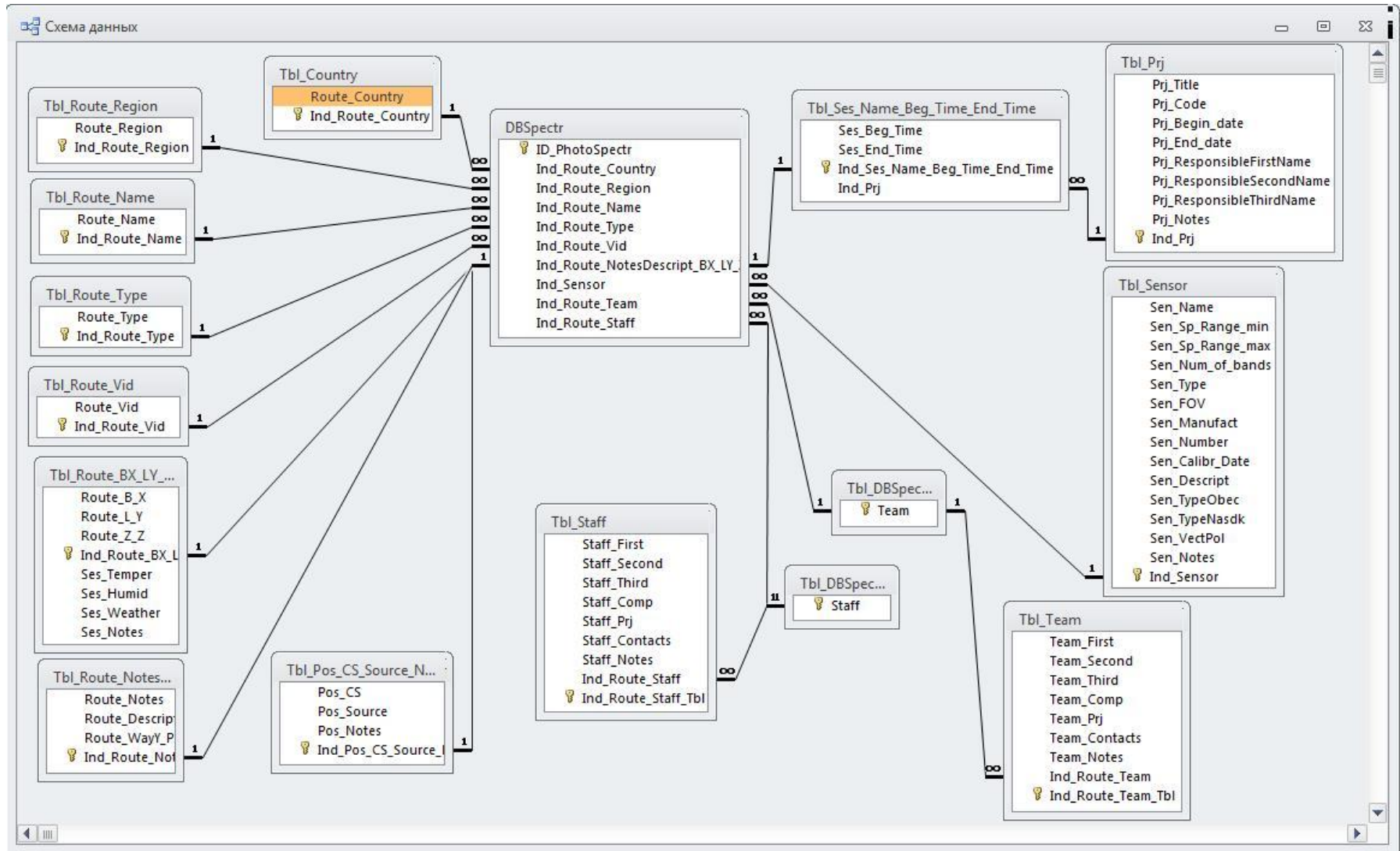


# Вопросы по ООП и С#

# Школа «Молодого программиста»

Работа с базой данных в  
приложении ASP.NET MVC

# Реляционные базы данных



# Способы работы с базой данных на основе ADO.NET

- «Низкий» уровень – SqlConnection, SqlCommand, Transaction, SqlDataReader
- «Средний» уровень – DataSet, DataTable, DataAdapter
- «Высокий» уровень – ORM-библиотеки NHibernate, Entity Framework и другие

# Платформа Entity Framework

это объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для **.NET Framework** от Microsoft.

Где можно почитать подробнее:

<https://msdn.microsoft.com/ru-ru/data/ee712907>



# Entity Framework



# Преимущества Entity Framework

1. Маппинг
2. Отслеживание изменений в контексте и сохранение этих изменений в базе данных
3. «Быстрая» настройка работы с базой данных
4. Оперирование классами при работе с базой данных

# Недостатки Entity Framework

1. Производительность приложения
2. Производительность приложения
3. Производительность приложения

# Использование Entity Framework

**Благодарю за внимание**

Вопросы