

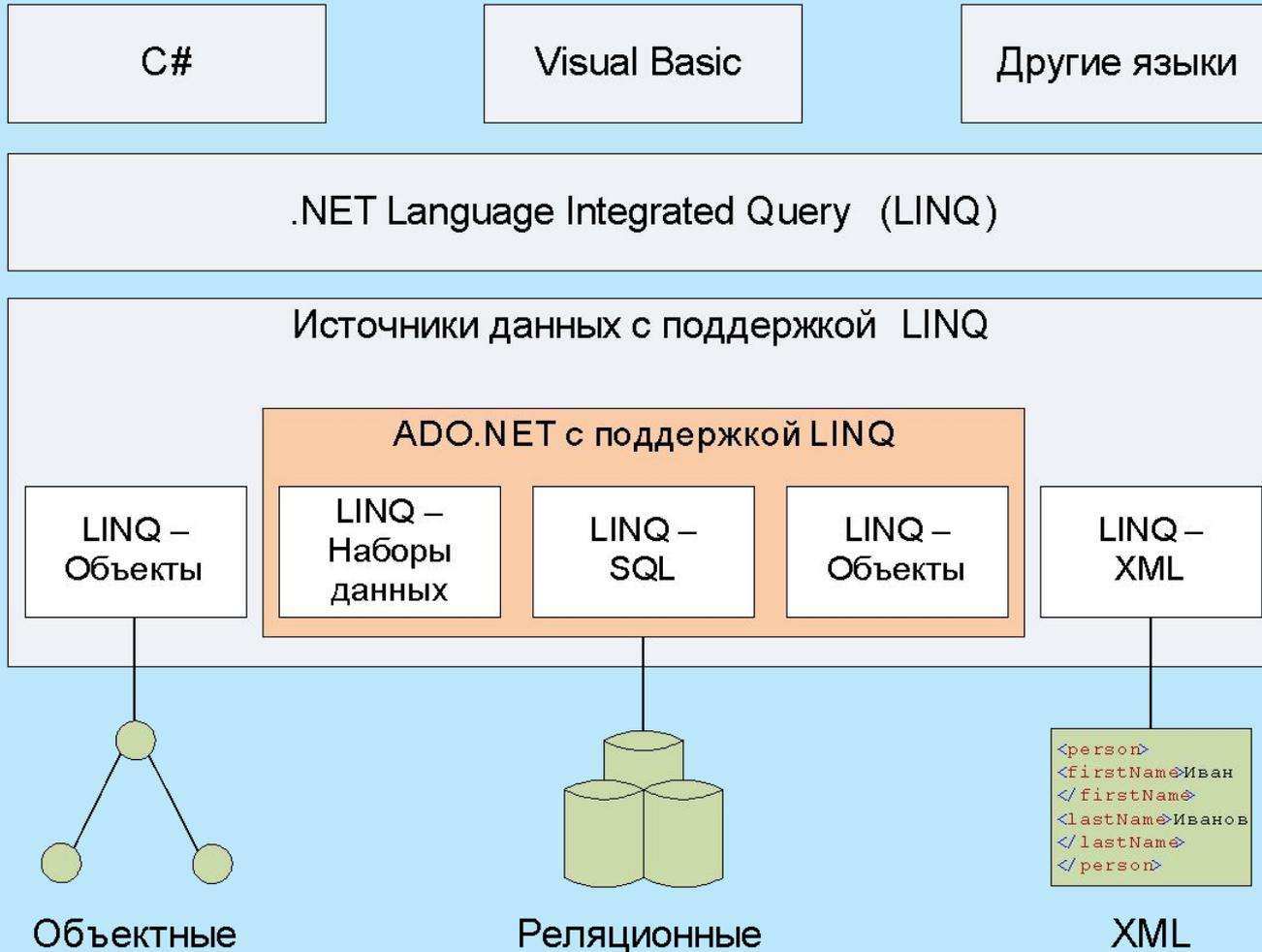
Подробный курс

Современные веб-технологии

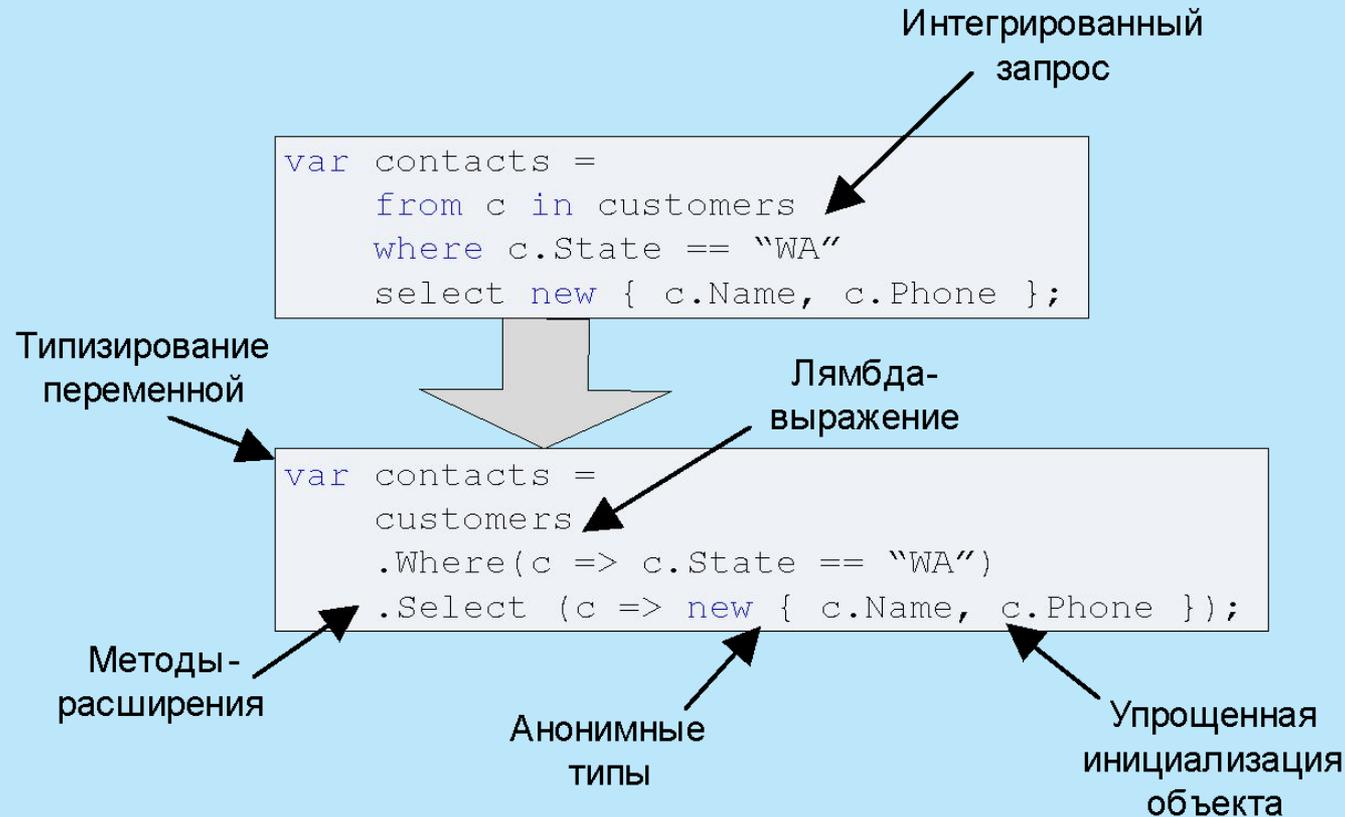
Проектирование баз данных и
работа с ними Веб-приложений.
LINQ, ADO.NET Entities, DDD

- **Language Integrated Query (LINQ)** – проект Microsoft по добавлению синтаксиса языка запросов, напоминающего SQL, в языки программирования платформы .NET Framework
- LINQ выпущен вместе с Visual Studio 2008 в конце ноября 2007 года.
- Запрашиваемые данные могут быть представлены в форме:
 - XML (запросы LINQ к XML)
 - баз данных (LINQ к SQL, LINQ к наборам данных и LINQ к экземплярам)
 - объектов (LINQ к объектам) и т.д.

Архитектура LINQ



Примеры написания запроса LINQ на языке C#



Операция запроса в LINQ



Источник: Введение в запросы LINQ ,

<http://msdn.microsoft.com/ru-ru/library/bb397926.aspx>

LINQ to SQL

- LINQ to SQL – простая, но достаточно мощная система объектно-реляционного отображения (ORM)
- LINQ to SQL позволяет производить отображение реляционных данных на объектную модель
- LINQ to SQL поддерживает все основные возможности, необходимые для разработчиков, на SQL:
 - Выбор
 - Вставка
 - Обновление
 - Удаление

Работа с реляционными данными по технологии ADO.NET

```
SqlConnection c = new SqlConnection(...);
c.Open();
SqlCommand cmd = new SqlCommand(
    "SELECT c.Name, c.Phone
    FROM Customers c
    WHERE c.City = @p0");
cmd.Parameters.AddWithValue("@p0", "London");
DataReader dr = c.Execute(cmd);
while (dr.Read()) {
    string name = dr.GetString(0);
    string phone = dr.GetString(1);
    DateTime date = dr.GetDateTime(2);
}
dr.Close();
```

Запрос в
кавычках

Аргументы без
строгих границ

Нетипизированные
возвращаемые
значения

Нет ошибок
времени
компиляции

Работа с реляционными данными по технологии LINQ to SQL

```
public class Customer { ... }  
  
public class Northwind : DataContext  
{  
    public Table<Customer> Customers;  
    ...  
}
```

Объектное
описание
данных

Таблицы описаны
как коллекции

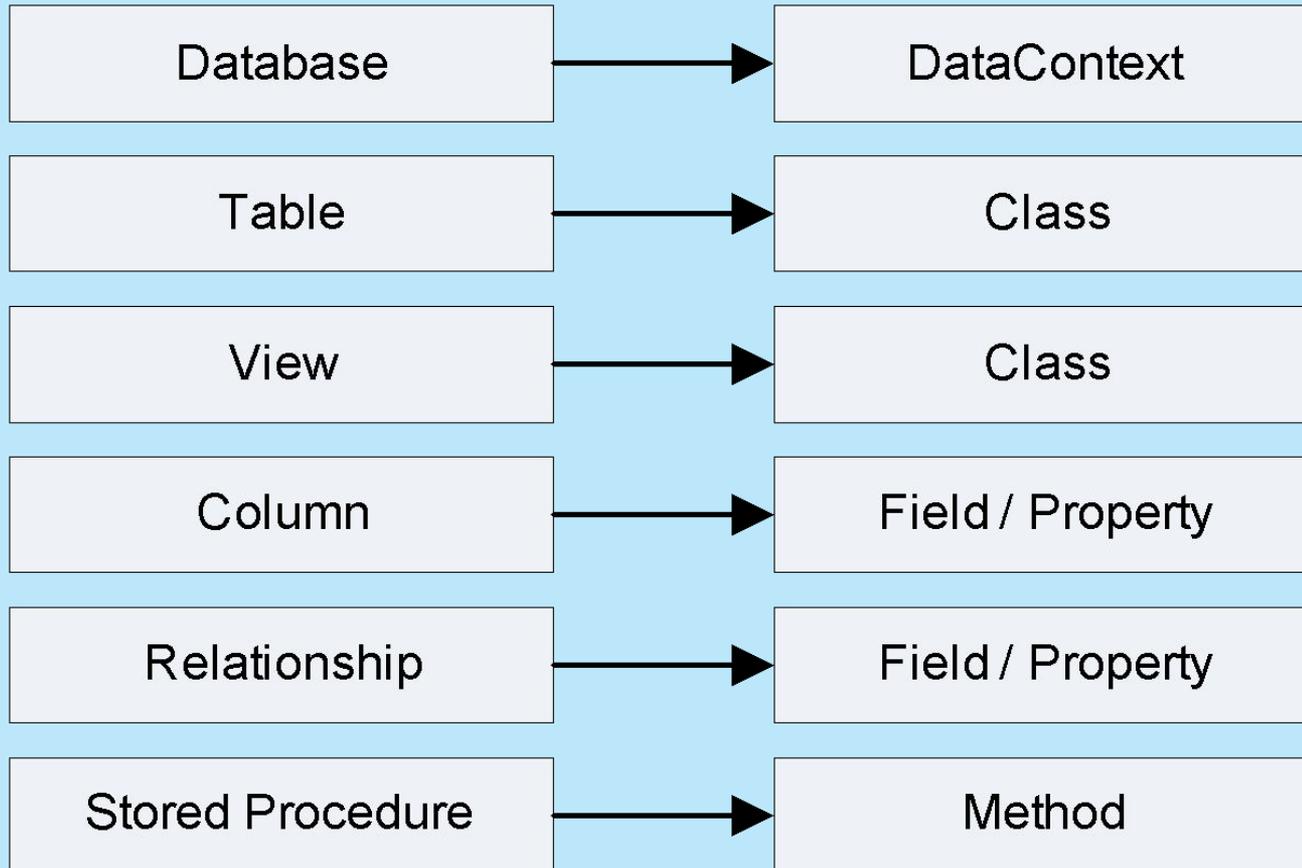
```
Northwind db = new Northwind(...);  
var contacts =  
    from c in db.Customers  
    where c.City = "London"  
    select new { c.Name, c.Phone };
```

Строго типизированное
соединение

Интегрированный
синтаксис запросов

Строго типизированные
возвращаемые значения

Отображение LINQ to SQL



ADO.NET Entity Framework

- **ADO.NET Entity Framework (EF)** – объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft
- Возможность взаимодействия посредством LINQ to Entities и с использованием Entity SQL
- Используется ADO.NET Data Services и связка из Windows Communication Foundation и Windows Presentation Foundation

Преимущества приложений ADO.NET Entity Framework

- Могут работать концептуальной моделью в терминах предметной области
- Освобождаются от жестких зависимостей от конкретного ядра СУБД или схемы хранения
- Сопоставления между концептуальной моделью и схемой, специфичной для конкретного хранилища
- Возможность работы с согласованной моделью объектов приложения, которая может быть сопоставлена с различными схемами хранения
- Несколько концептуальных моделей могут быть сопоставлены с единой схемой хранения
- Поддержка LINQ

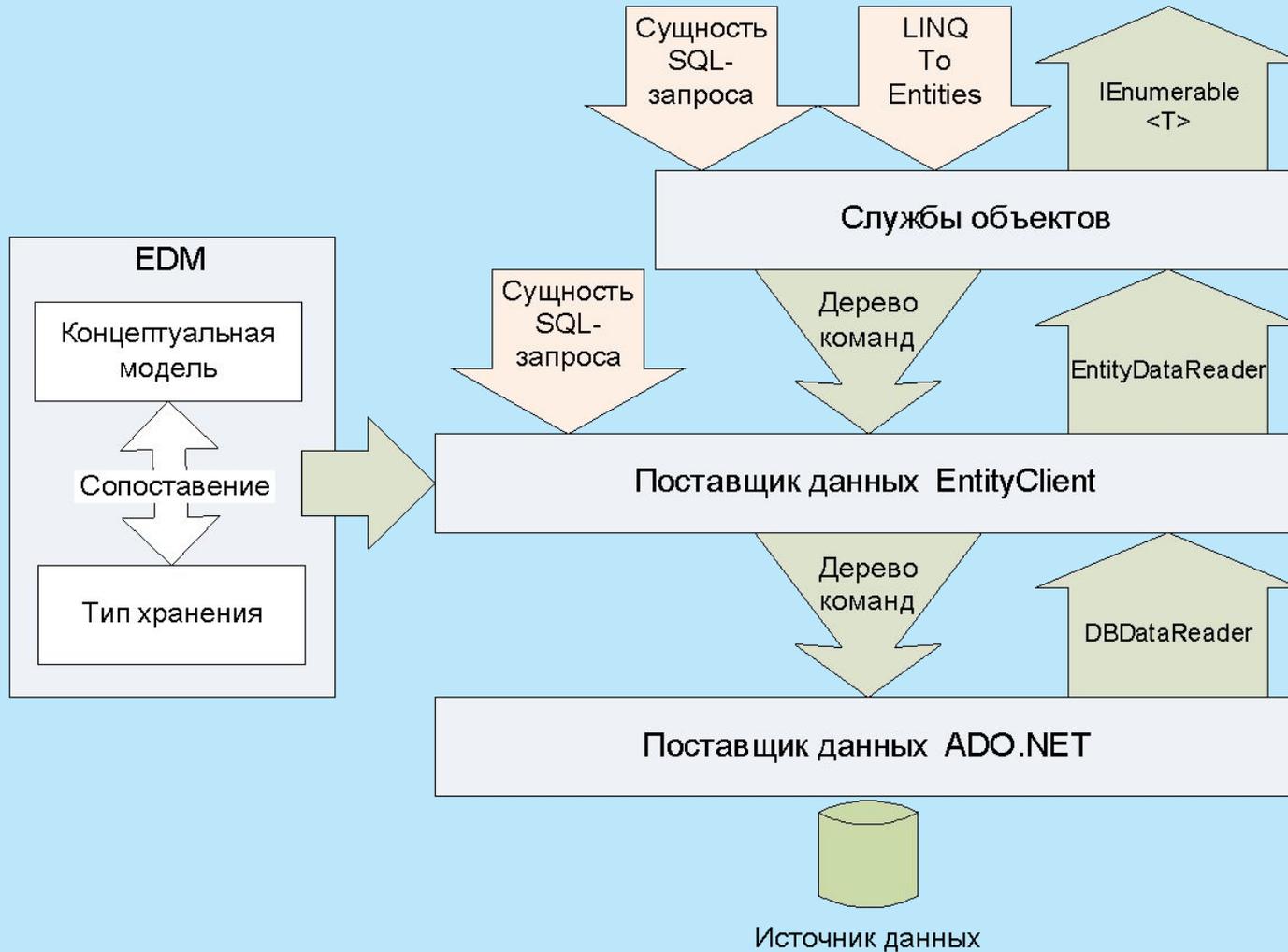
Компоненты Entity Framework

- Модель Entity Data Model (EDM)
- Компонент Object Services
- Компонент LINQ to Entities
- Язык Entity SQL
- Поставщик EntityClient
- Компонент метаданных ADO.NET
- Набор средств, которые создают сопоставления и разделяемые классы, представляющие сущности концептуальной модели
- Поставщик данных SqlClient

Entity Data Model



Архитектура Entity Framework



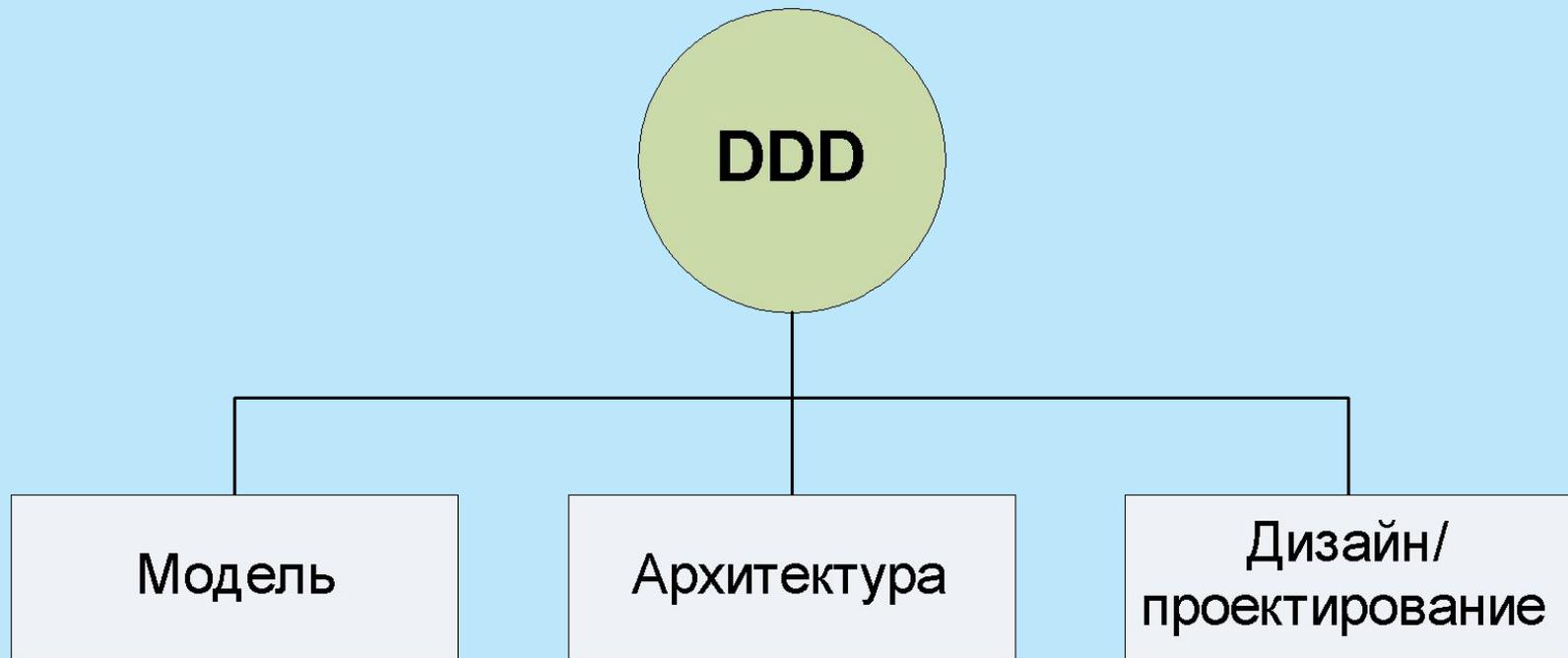
Проблемно-ориентированное проектирование

- **Проблемно-ориентированное проектирование (Domain-Driven Design, DDD)** является подходом к разработке программного обеспечения для решения сложных задач, глубоко связывающем реализацию программной систем с развивающейся моделью бизнес-концепций
- **Свойства DDD:**
 - фокусировка проекта на основной проблемной области (домене) и бизнес-логике этой области
 - базирование всего проектирования на модели
 - введение творческого сотрудничества между техническими экспертами и экспертами в проблемной области, чтобы максимально понять концептуальную суть проблемы

Основные понятия DDD

- Домен (проблемная область)
- Модель
- Общеупотребительный язык (Ubiquitous Language)
- Контекст

Основные аспекты DDD



Архитектурный аспект DDD

- Сущность (Entity)
- Объект со значением (Value Object)
- Агрегат (Aggregate)
- Службы (Services)
- Репозитории (Repositories)
- Фабрики (Factories)