ЛАБОРАТОРНАЯ РАБОТА №2

АРХИТЕКТУРА ПО.

ENTITYFRAMEWORK. LINQ

Содержание работы:

Необходимо реализовать работу с БД в приложении, используя ORM Entity Framework. За основу взять WinForm приложение из предыдущей л.р.

Часть 1. Взаимодействие с БД с помощью ORM EntityFramework.

1. Использовать подход Database-first или Code-second для существующей БД.

Реализовать CRUD для сущности.

1. Изменить класс сущности (например тип, название свойства, указать точность и т.п.) и сконфигурировать EF (использовать Аннотации метаданных и Fluent API)
2. Изменить класс(ы) сущности (например добавить или удалить свойство/класс) и выполнить миграцию БД
3. Добавить на форму выполнение выборки с использованием хранимой процедуры.
4. Продемонстрировать работу приложения.

Часть 2. Работа с данными с использованием технологии LINQ.

1. Огранизовать выборку данных с помощью запроса LINQ. В запросе использовать объединение/группировку, проекцию, фильтрацию.

Часть 3. Использование EntityFramework для проектирования реляционной БД.

1. Создать новое приложение
2. Использовать подход Code-First для создания новой БД. Предметная область выбирается по желанию студента. Желательно попробовать данный подход для сущностей, связанных отношением наследования. Минимальное количество сущностных классов – два.
3. Продемонстрировать созданную БД.

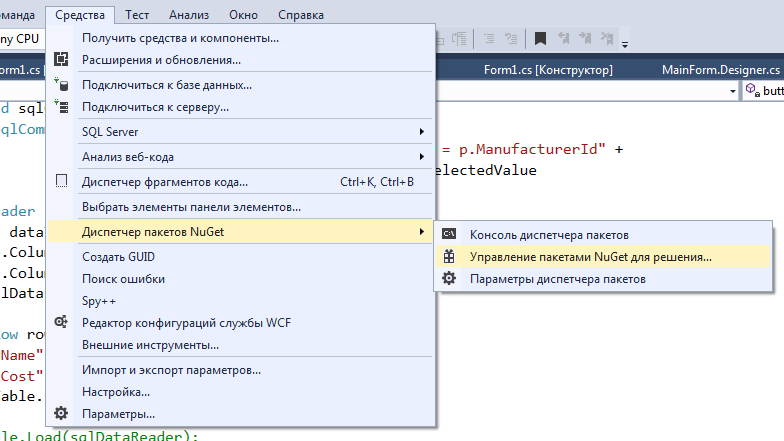
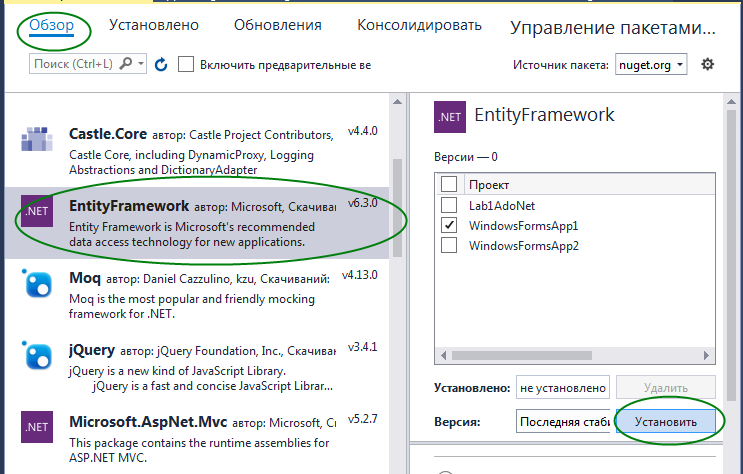
Вопросы для защиты лаб. работы:

1. Методы Linq
2. Настройка модели EF с помощью аннотаций и fluent api
3. Связь сущностей в модели - аналог FK БД, виды наследования классов сущностей
4. Что будет, если в конструктор контекста не передавать аргумент
5. Продемонстрировать механизм миграции (проект code First)

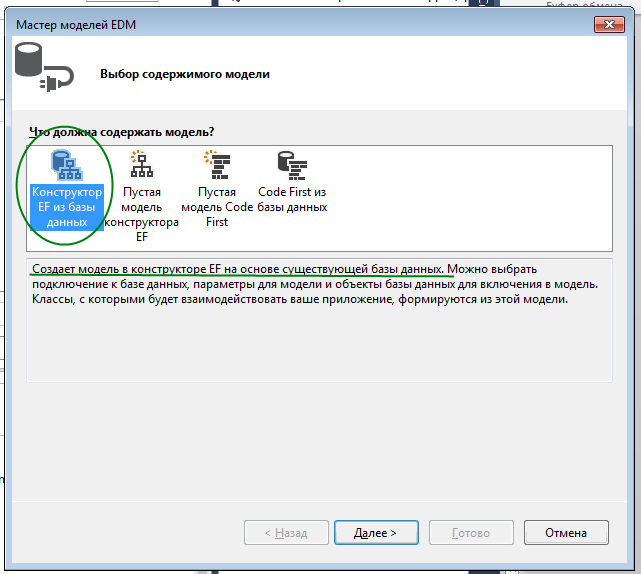
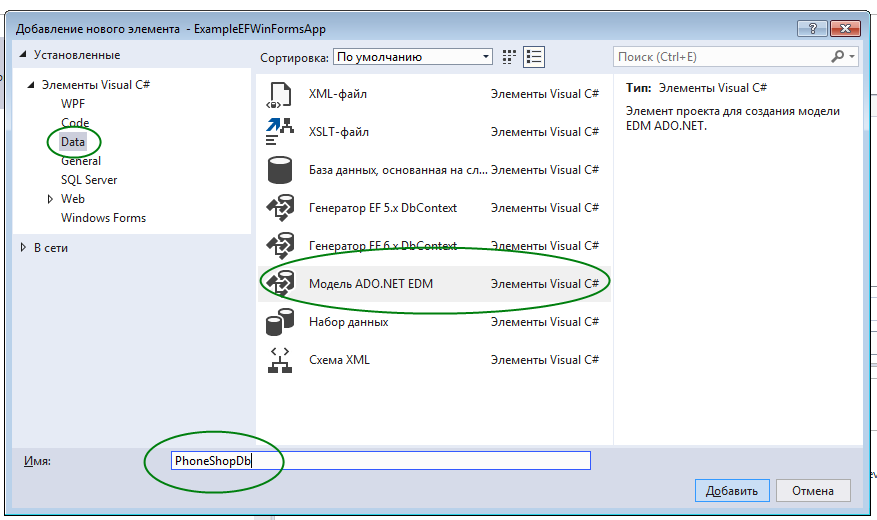
Основные этапы работы.

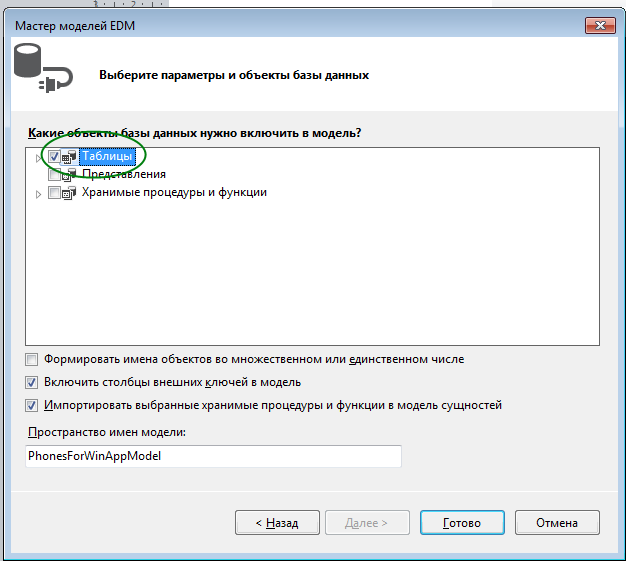
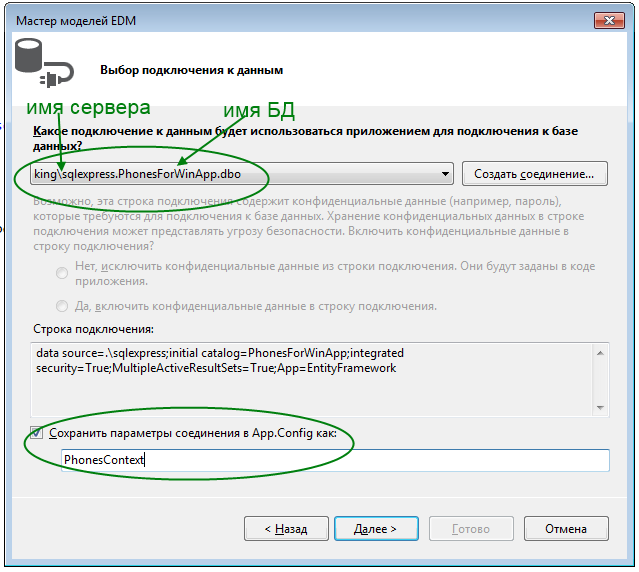
*Часть 1. Взаимодействие с БД с помощью ORM EntityFramework.*

1. Добавление бибилиотеки EF через пакетный менеджер Nuget

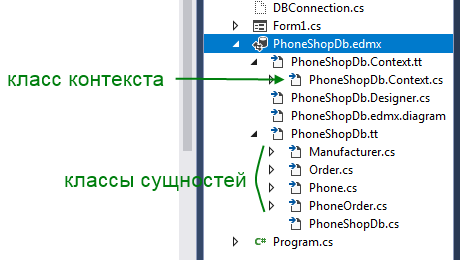


1. Генерация сущностных классов с помощью EF: добавить новый элемент “модель EDM” в проект, указав сервер БД и название БД, название для строки подключения, таблицы БД для модели EDM.

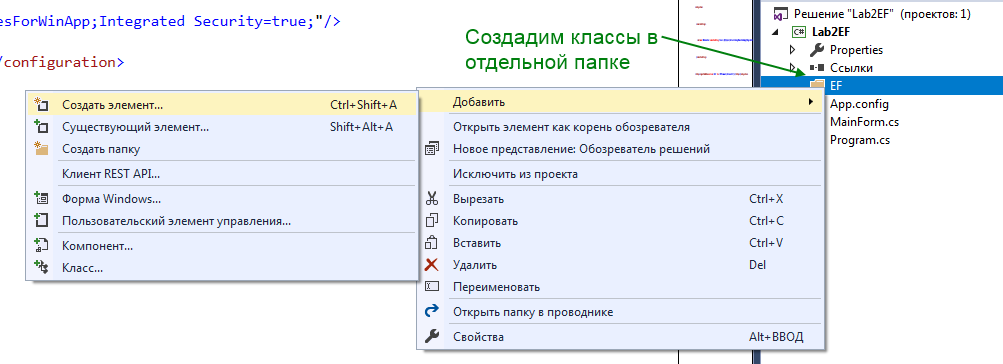


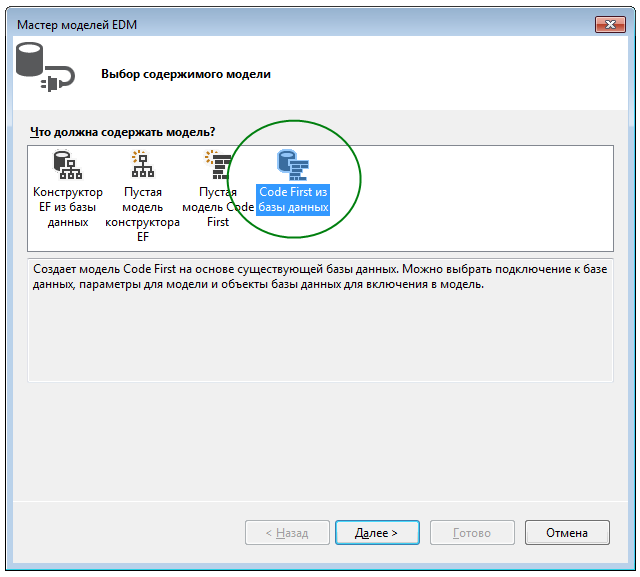
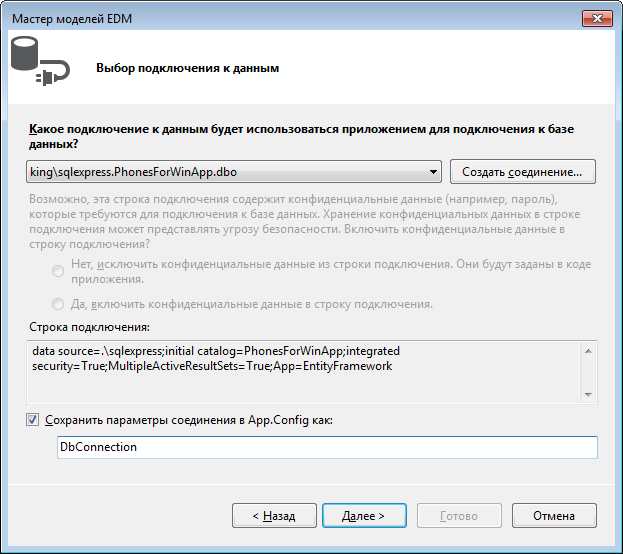


В итоге в проект будет добавлен файл модели .edmx, класс контекста и классы сущностей для выбранных таблиц.

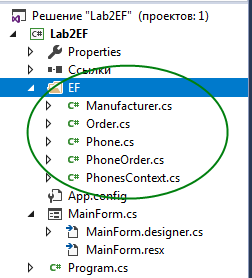


Если модель edmx не нужна, можно воспользоваться вариантом Code-Second. Добавление в проект только файлов контекста и сущностей:

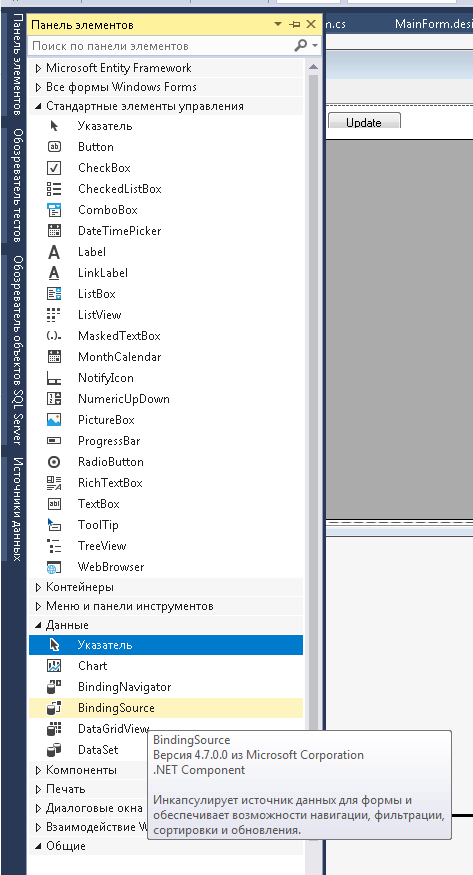
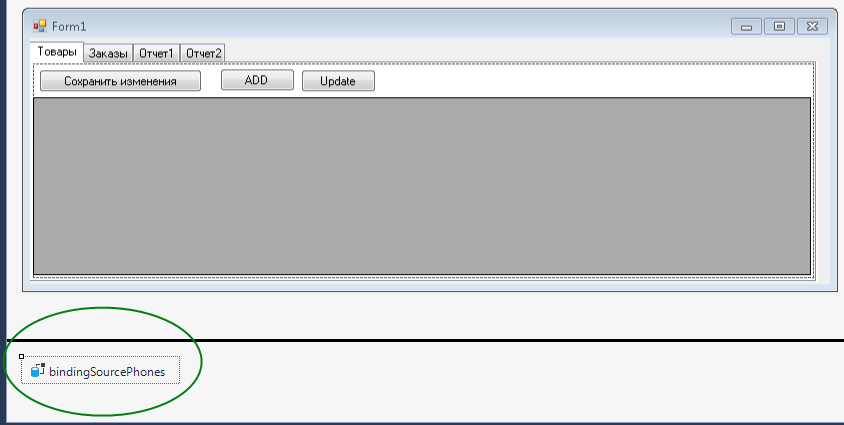


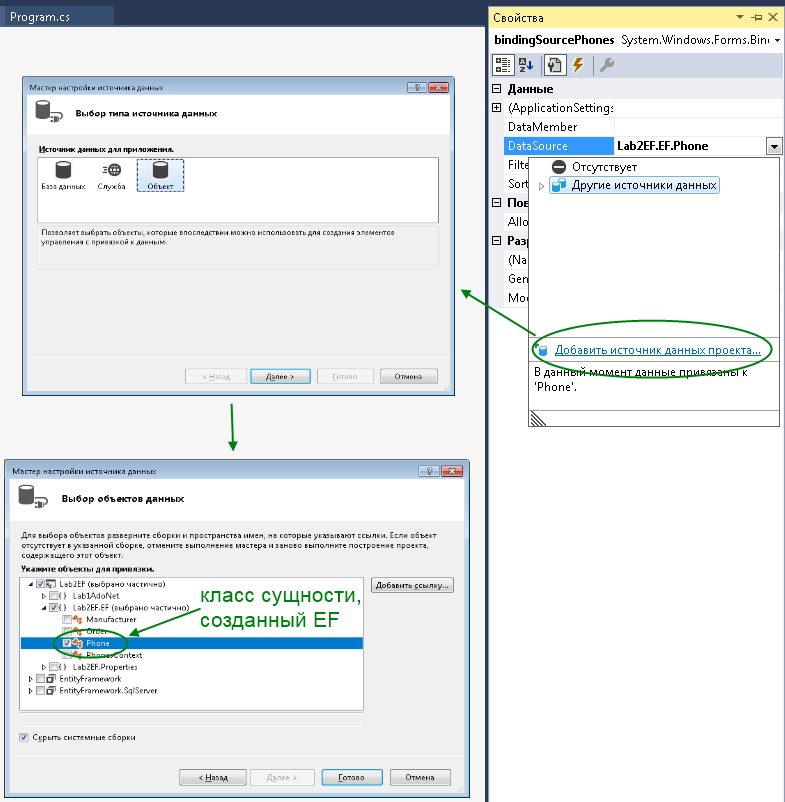
 

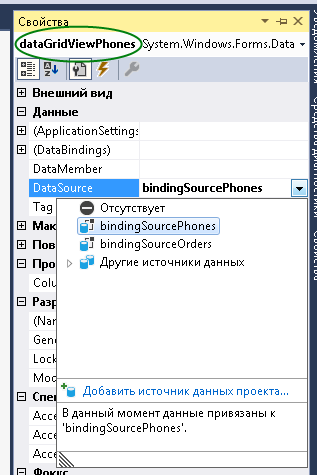
В итоге в папке EF созданы только контекст и сущности:

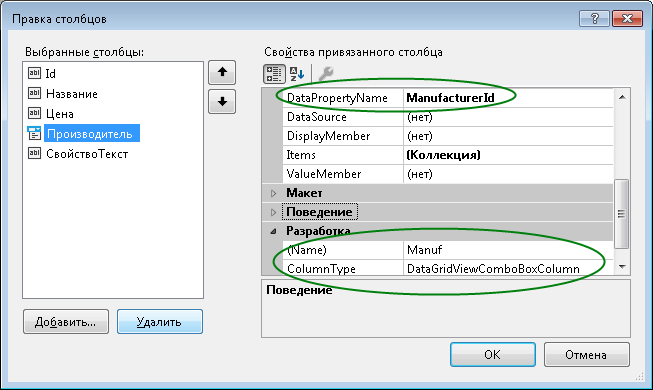


1. Для отображения данных в таблице можно воспользоваться элементом BindingSource. Для него настроим в качестве источника данных соответствующий сущностный класс. В данном примере это сущность Phone. Настроенный элемент BindingSource будет служить источник данных таблицы DataGrid. В результате в таблице DataGrid автоматически добавятся все свойства указанной сущности в виде столбцов. Свойство - ссылку на справочник изменим на выпадающий список аналогично предыдущей л.р. Чтобы в элементе BindingSource были доступны для выбора классы сущностей, необходимо проект пересобрать.

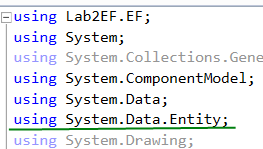
 



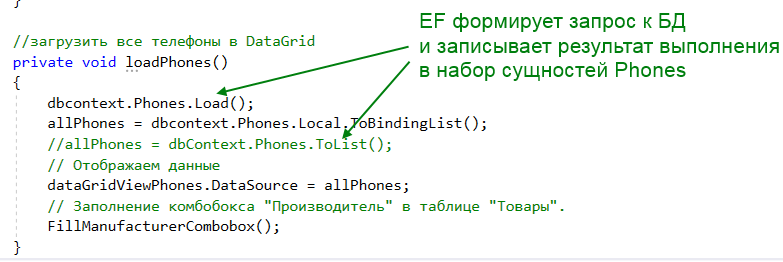




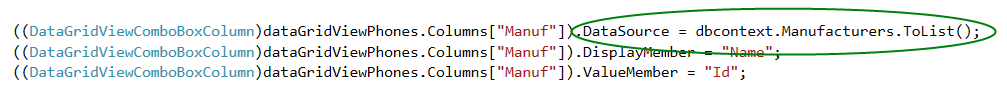
1. Для обращения к таблицам БД из кода потребуется объект контекста данных. Например, телефоны находятся в объекте Phones контекста данных. Для хранения данных в оперативной памяти используется класс List или BindingList.



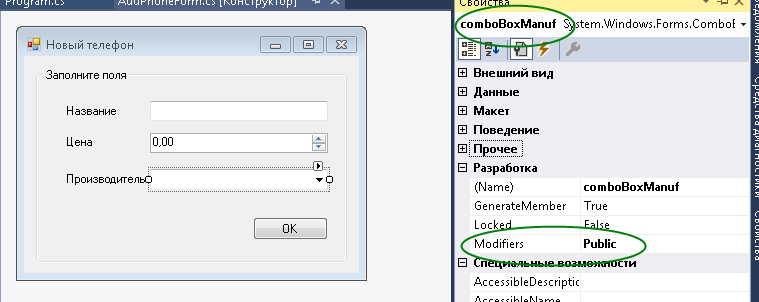




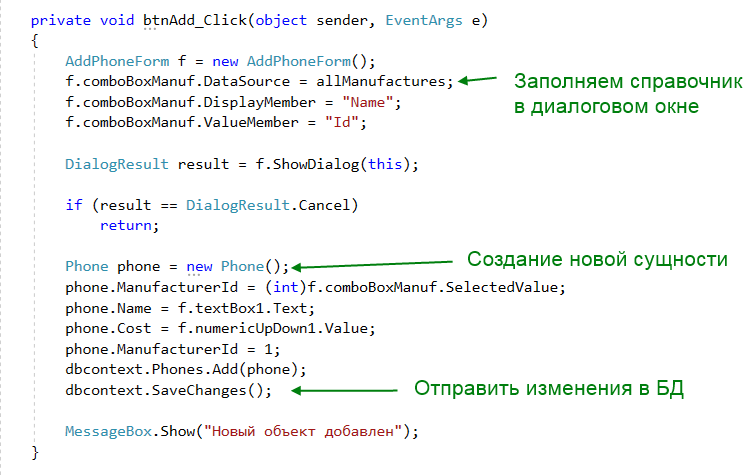
1. Выпадающий список справочника заполняется аналогично, обращаясь к набору соответствующих сущностей в контексте данных.

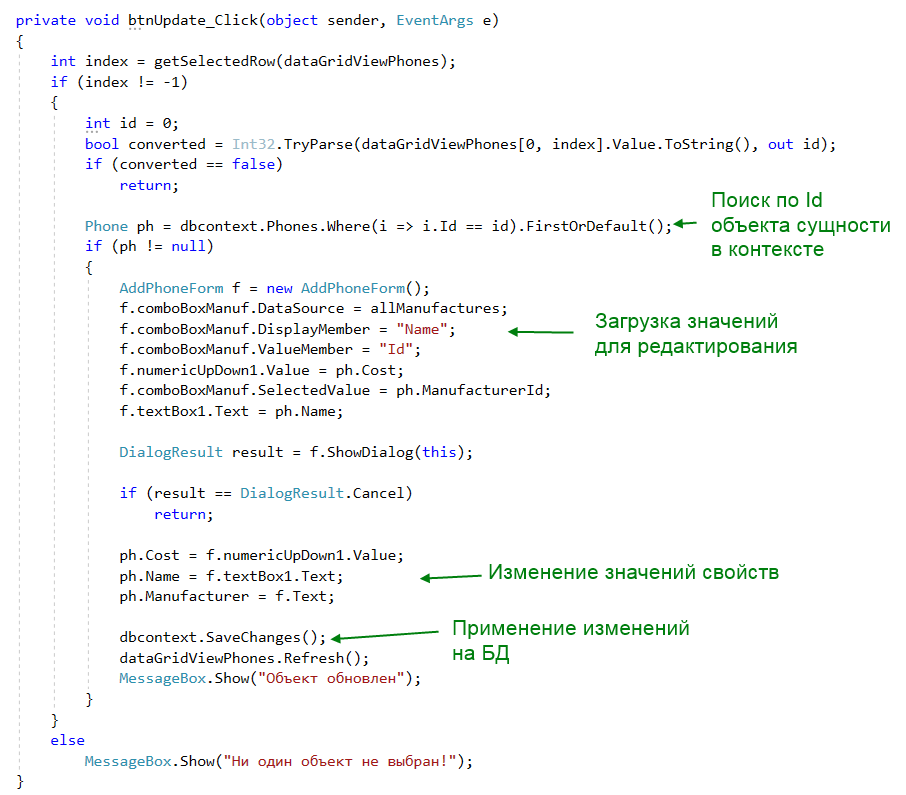


1. Для применения изменений на базу данных необходимо вызвать метод контекста данных SaveChanges()
2. Добавление и редактирование данных с использованием формы редактирования сущности. Можно обращаться к элементам диалогового окна из главного, установив модификатор доступа элементов public.



Для добавления новой записи в БД необходимо создать экземпляр сущности, задав значения свойств, и вызвать метод SaveChanges() для генерации и отправки sql-команды в СУБД. При изменении существующей в БД записи, найдем ее в контексте (например, по идентификатору), изменим значения свойств и вызовем метод SaveChanges().





1. Для выполнения миграций выполняются следующие шаги:

Package Manager Console:

enable-migrations (добавление функционала миграций)

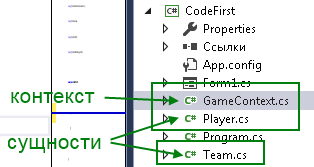
add-migration "название\_миграции" ( в папку Migrations будет добавлена новая миграция)

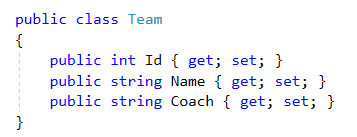
update-database (выполнить миграцию обновить БД)

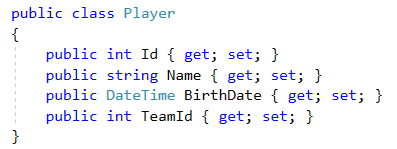
1. Вызов хранимой процедуры с помощью метода SqlQuery<T>()

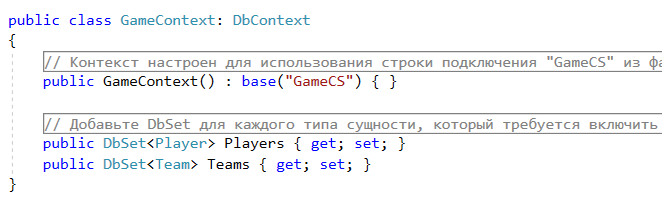
*Часть 3. Использование EntityFramework для проектирования реляционной БД*.

1. Создать новый проект WinForms
2. Подключить EF, если она не была подключена ранее на решение.
3. Определить базовые классы сущностей C# и класс контекста данных.

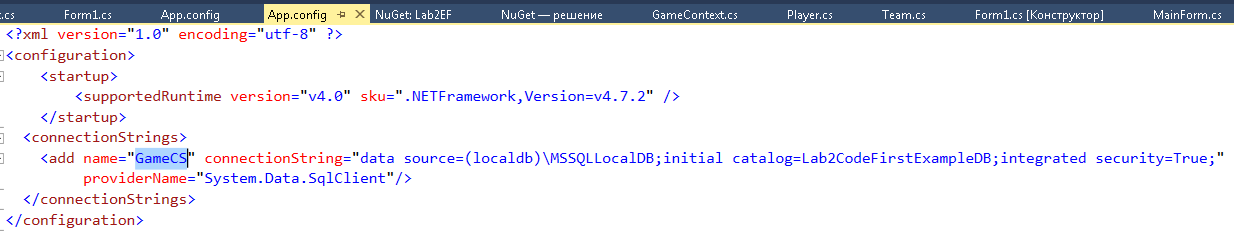




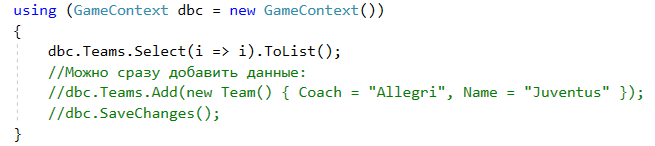




1. Задать строку подключения в конфиге с именем, указанным в классе контекста. Для подхода Code First EF требуются сведения о провайдере.



1. Запустить код взаимодействия с БД: например, запрос или добавление данных.



1. Проверить создалась ли БД.