

Рисунок 1 – В первую очередь создаем проект консольное приложение **МАЙКРОСОФТ**

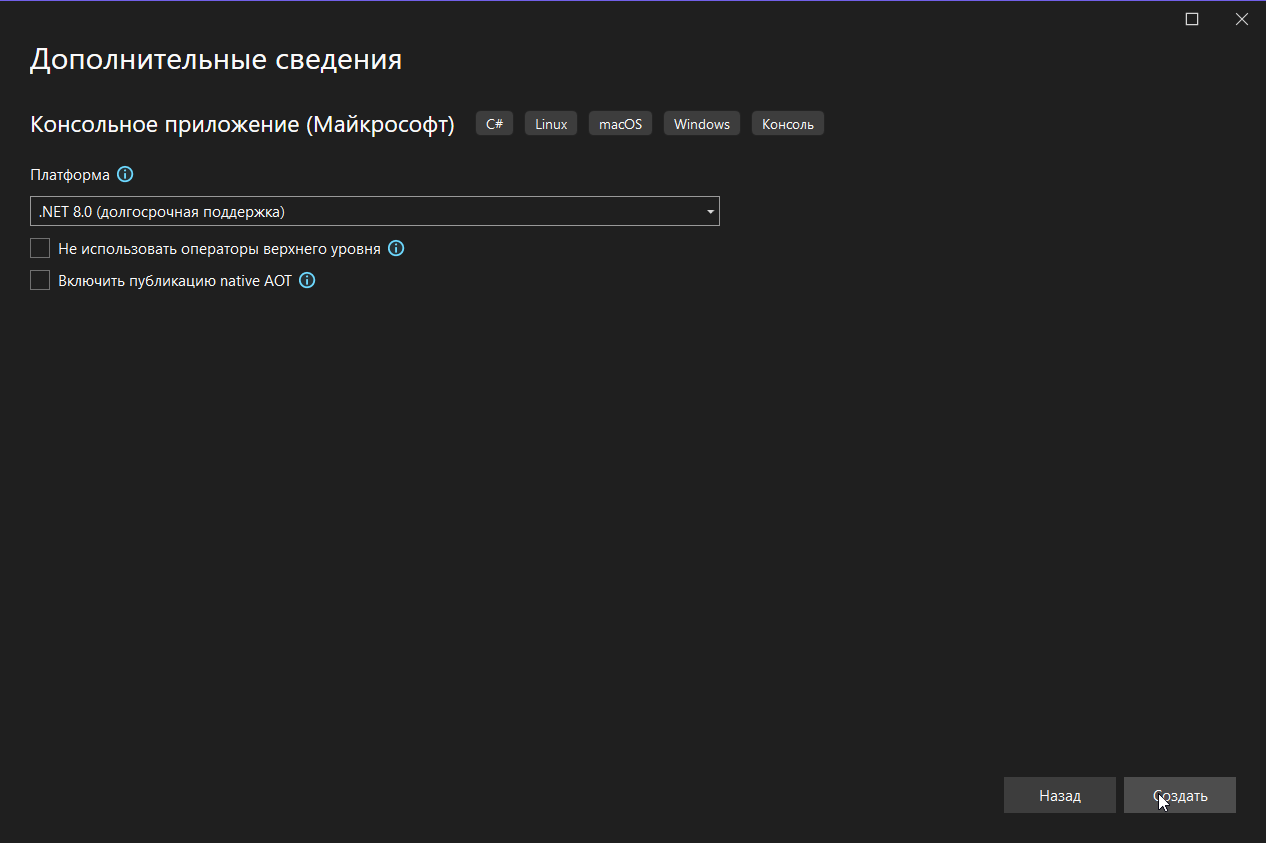


Рисунок 2 – при создании смотрите, чтобы версия .NET была 6.\*, 7.\* или 8.\*, но никак не 4.\*.\*

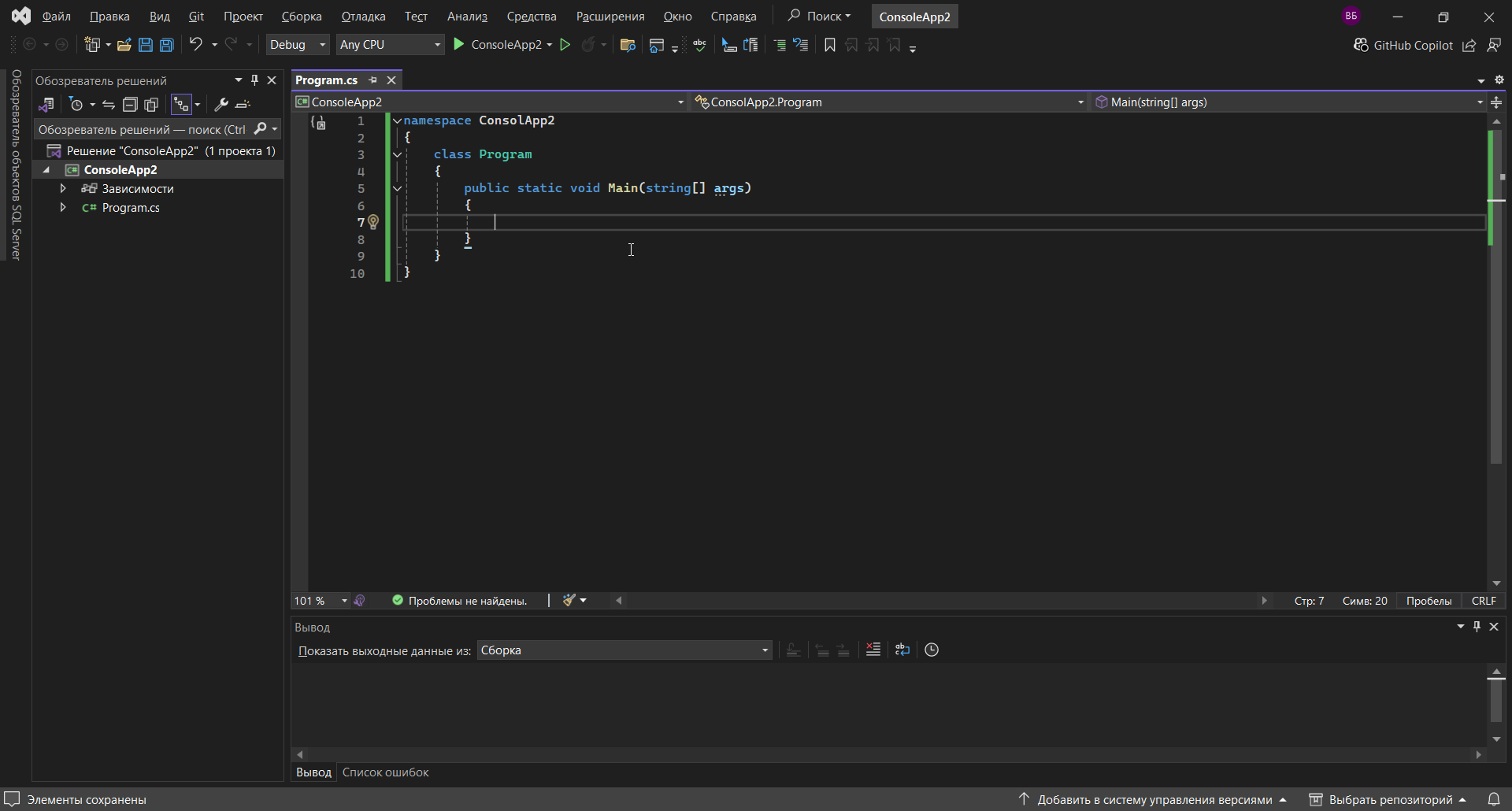


Рисунок 3 – создали базовую структуру консольного приложения

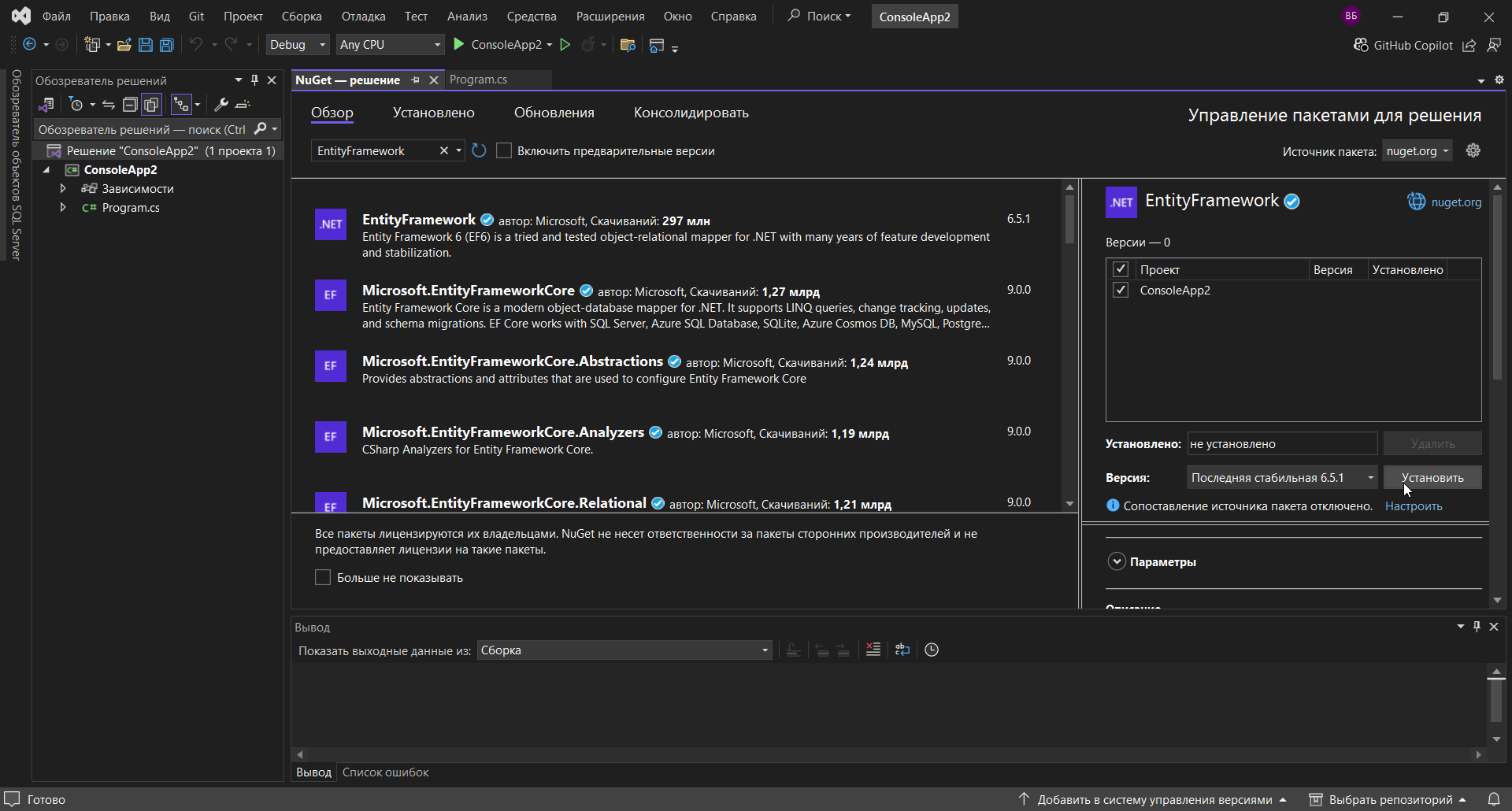


Рисунок 4 – устанавливаем EntityFramework, чтобы работать с базами данных

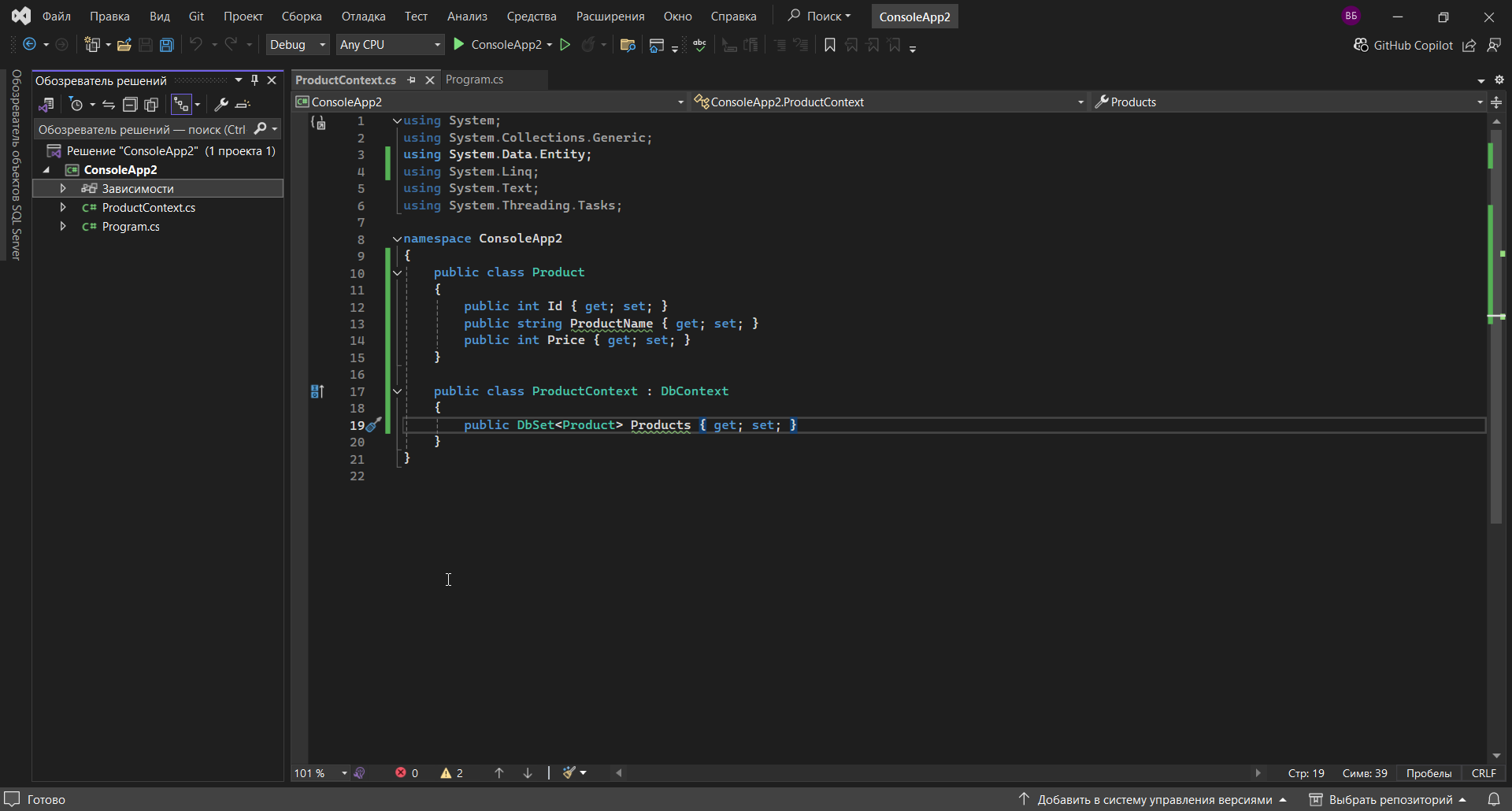


Рисунок 5 – создали файл ProductContext. Внутри него прописываем два класса : ProductContext (наследуется от DbContext) и Product (сущность или же таблица)

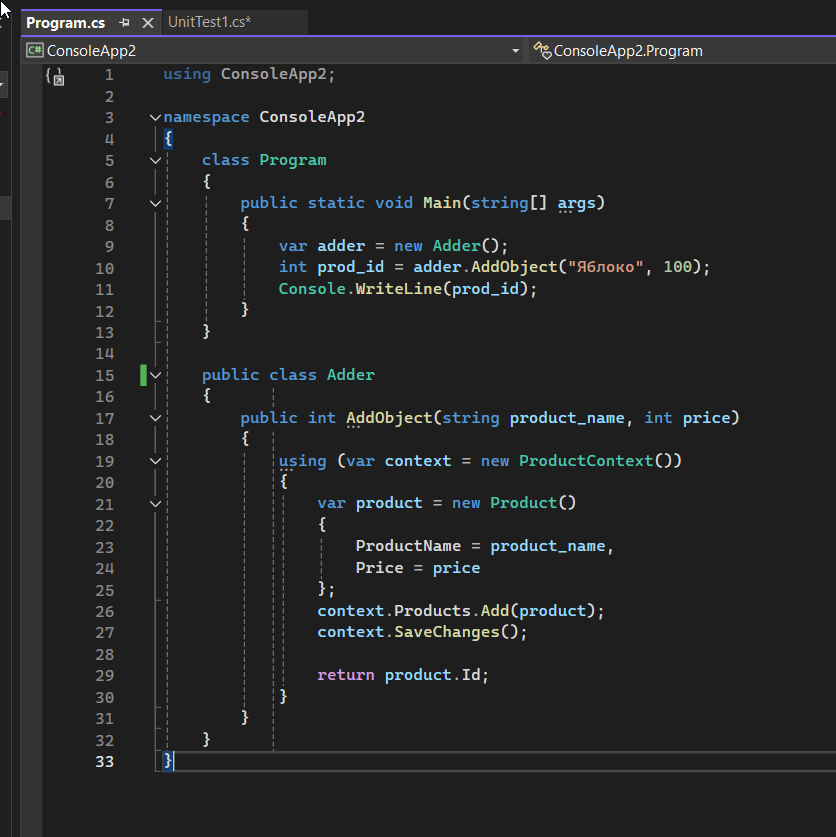


Рисунок 6 – в основном файле создаем с вами новый класс Adder, который будет добавлять новые объекты в таблицу Products (код в конце файла в листинге). Метод AddObject возвращает Id новосозданного продукта

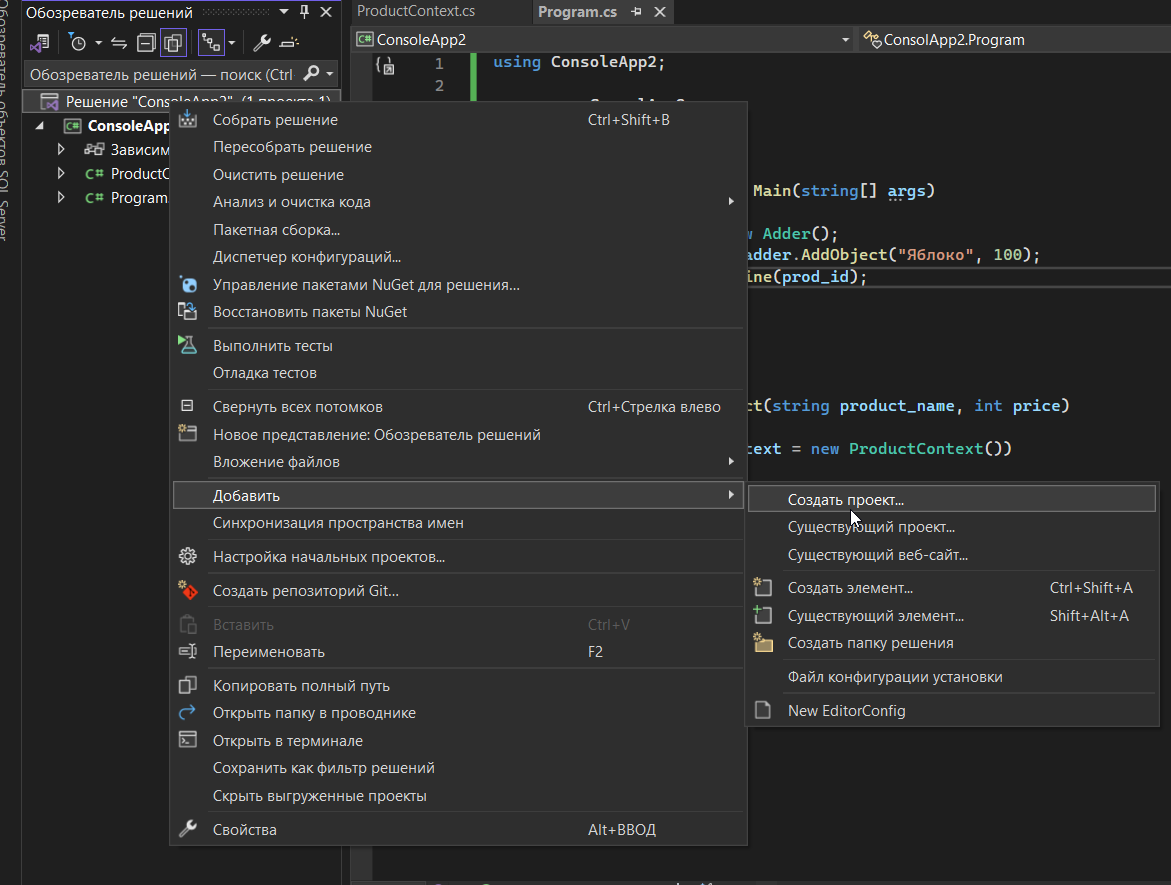


Рисунок 7 – далее мы нажимаем на наше **РЕШЕНИЕ** (фиолетовая иконка вижуалки), нажимаем добавить -> создать проект

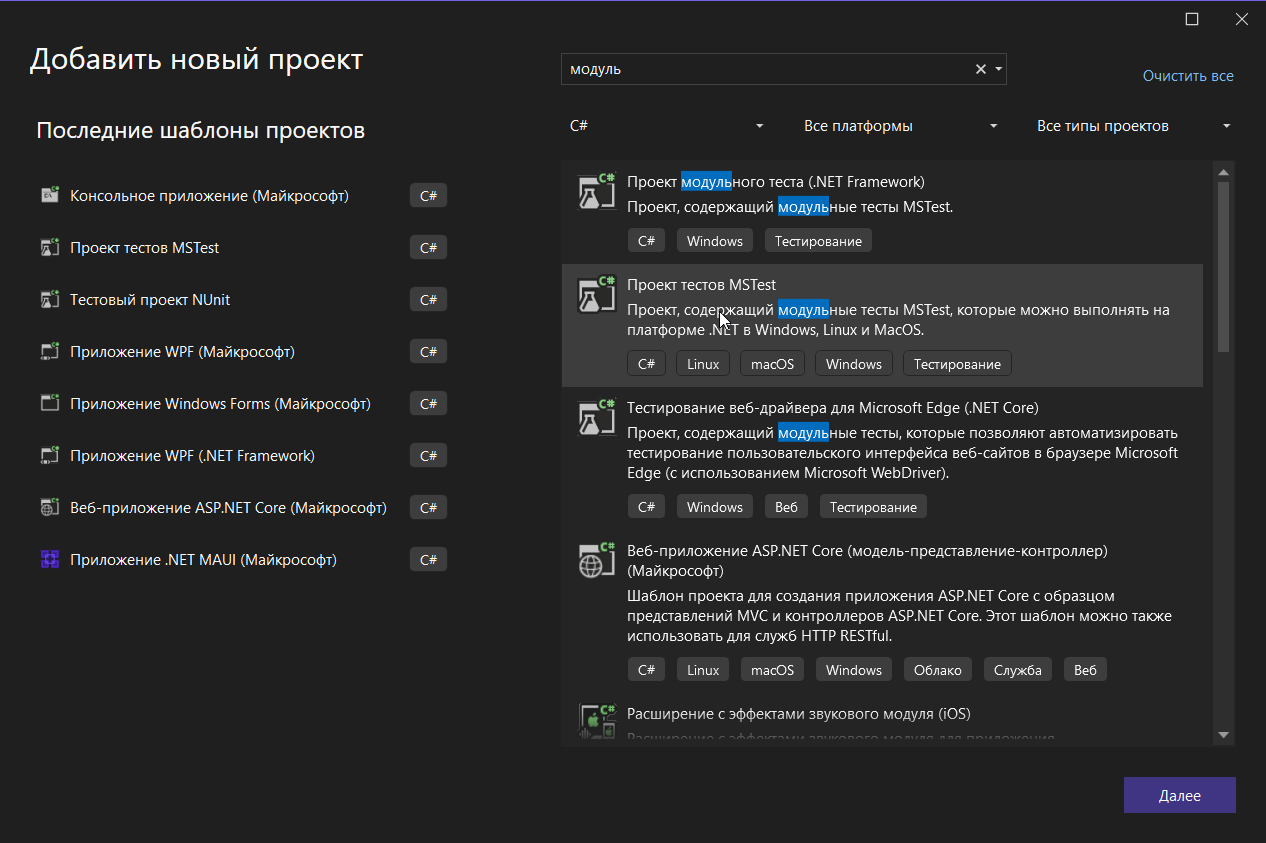


Рисунок 8 – Выбираем проект тестов **MSTEST** и никакой другой!!!!

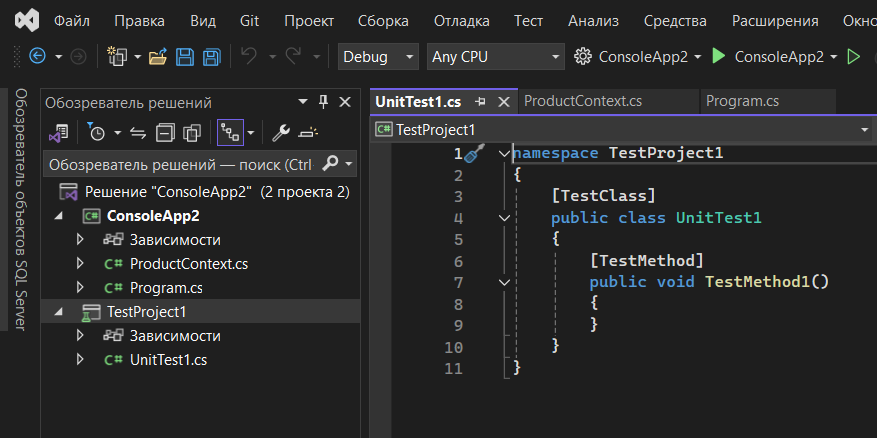


Рисунок 9 – вот так должно получиться

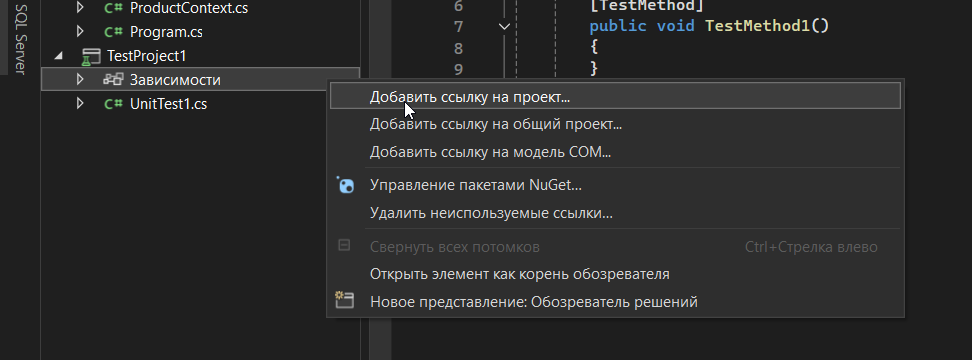


Рисунок 10 – нажимаем зависимости -> добавить ссылку на проект

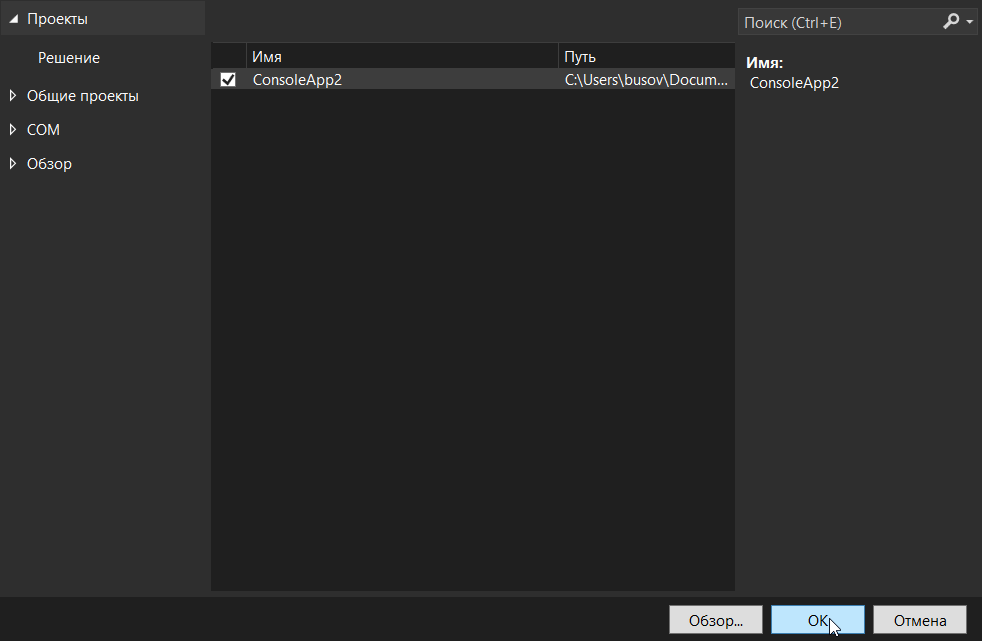


Рисунок 11 – выбираем тут наш проект

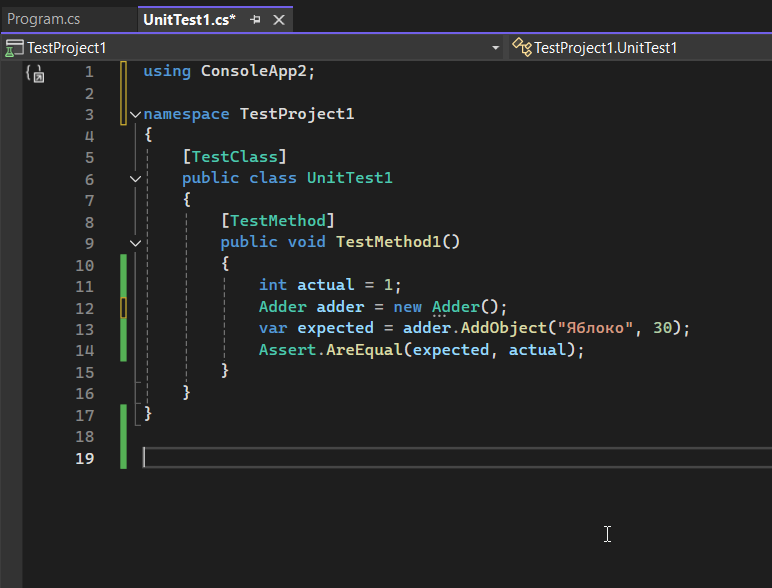


Рисунок 12 – Пишем вот такой вот код тестирования.  
**actual** – результат, который мы ожидаем получить.   
**expected** – то, что мы по факту получили, выполнив функцию.  
**Assert.AreEqual(expected, actual)** – сравнивает, одинаковый ли получился результат

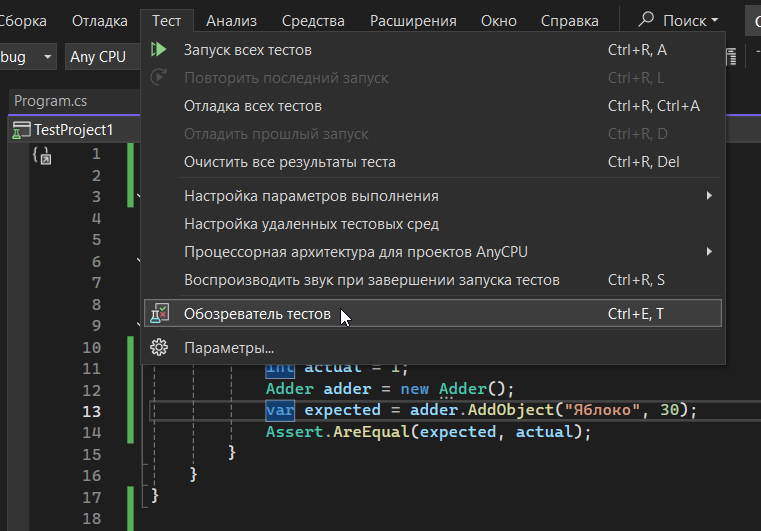


Рисунок 13 – нажимаем на «Тесты» -> «Обозреватель тестов»

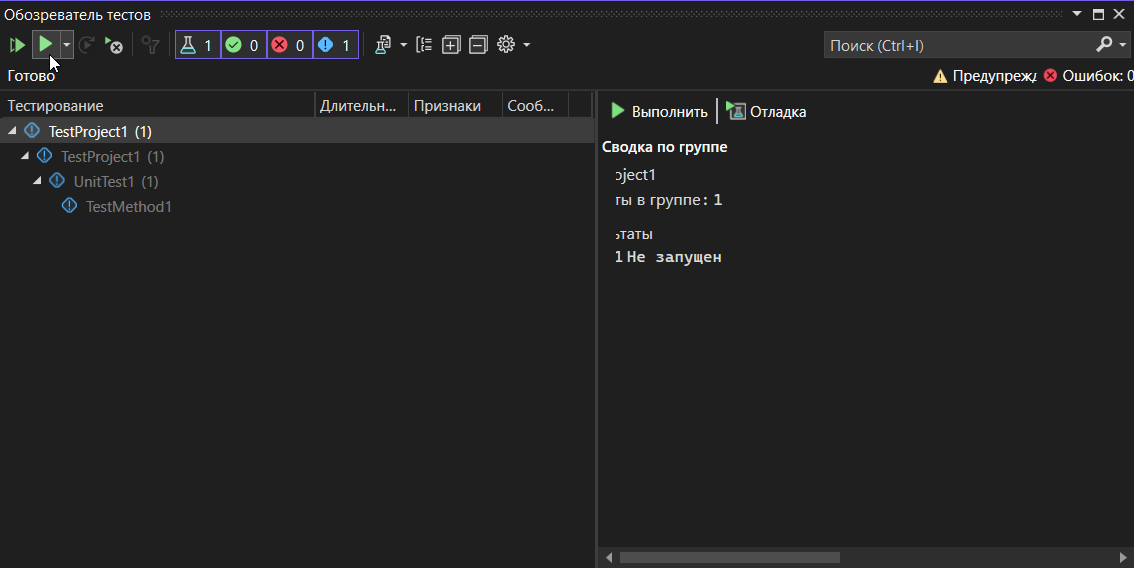


Рисунок 14 – выбираем наш тест и жмем зеленую стрелку

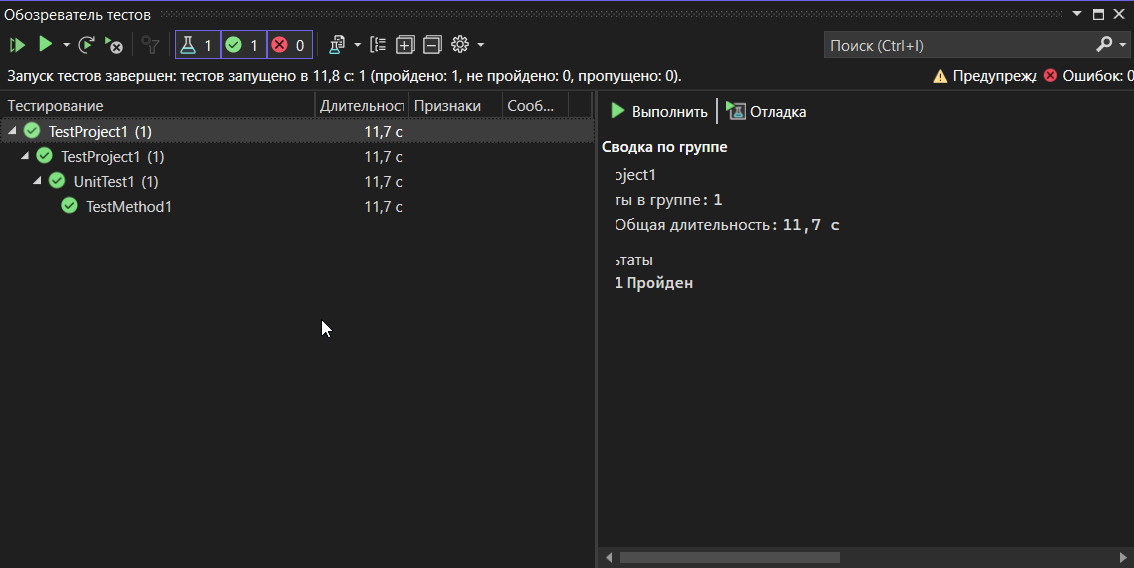


Рисунок 15 – результат теста – успех. Ожидаемое и фактические значения совпали

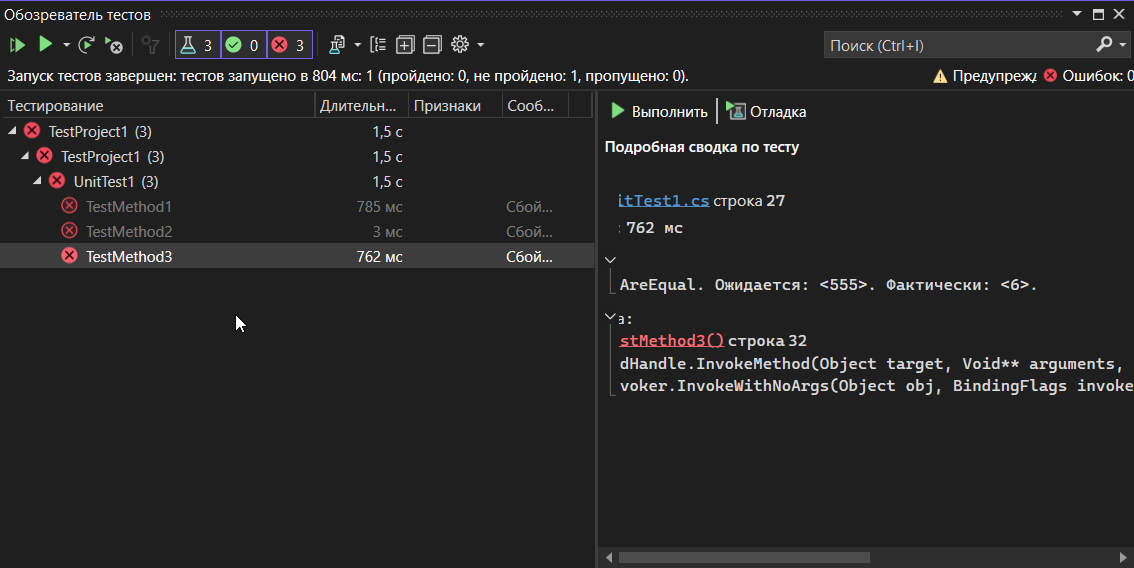


Рисунок 16 – так выглядит окно тестов, если фактическое и ожидаемые значения не совпали. Тут я добавил еще два теста, просто скопировав и вставив еще два раза функцию **public void TestMethod2** и **public void TestMethod3** соответственно

Вот так мы протестировали добавление нового объекта в базу данных.

**Листинг**

**UnitTest1.cs**

using ConsoleApp2;

namespace TestProject1

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod1()

{

int actual = 1;

Adder adder = new Adder();

var expected = adder.AddObject("Яблоко", 30);

Assert.AreEqual(expected, actual);

}

}

}

**Program.cs**

using ConsoleApp2;

namespace ConsoleApp2

{

class Program

{

public static void Main(string[] args)

{

var adder = new Adder();

int prod\_id = adder.AddObject("Яблоко", 100);

Console.WriteLine(prod\_id);

}

}

public class Adder

{

public int AddObject(string product\_name, int price)

{

using (var context = new ProductContext())

{

var product = new Product()

{

ProductName = product\_name,

Price = price

};

context.Products.Add(product);

context.SaveChanges();

return product.Id;

}

}

}

}

**ProductContext.cs**

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

public class Product

{

public int Id { get; set; }

public string ProductName { get; set; }

public int Price { get; set; }

}

public class ProductContext : DbContext

{

public DbSet<Product> Products { get; set; }

}

}