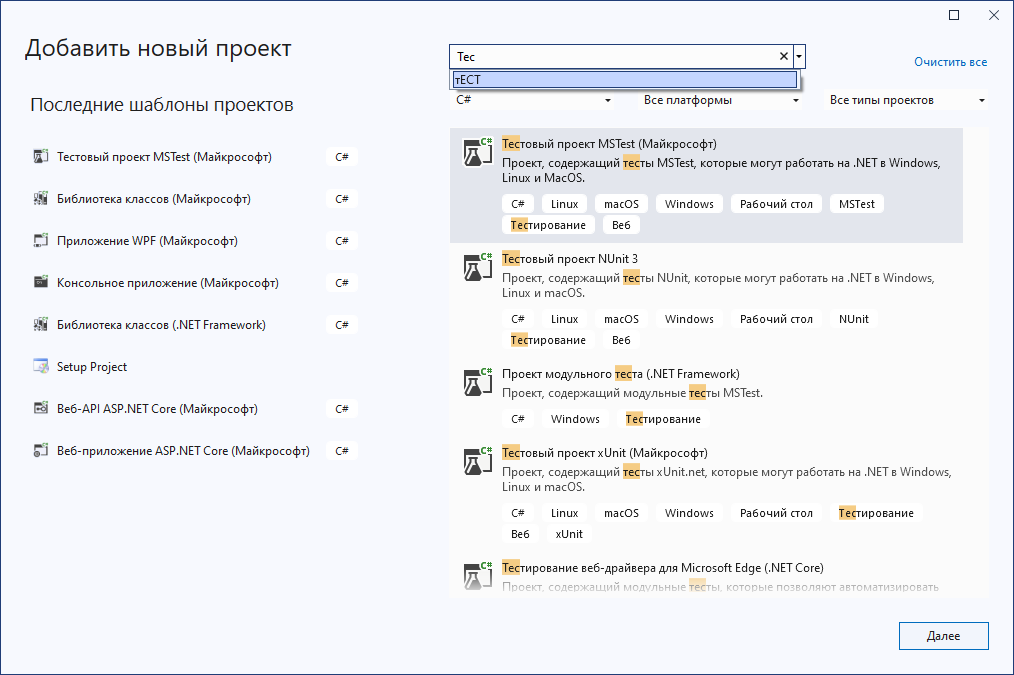
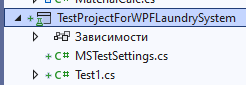
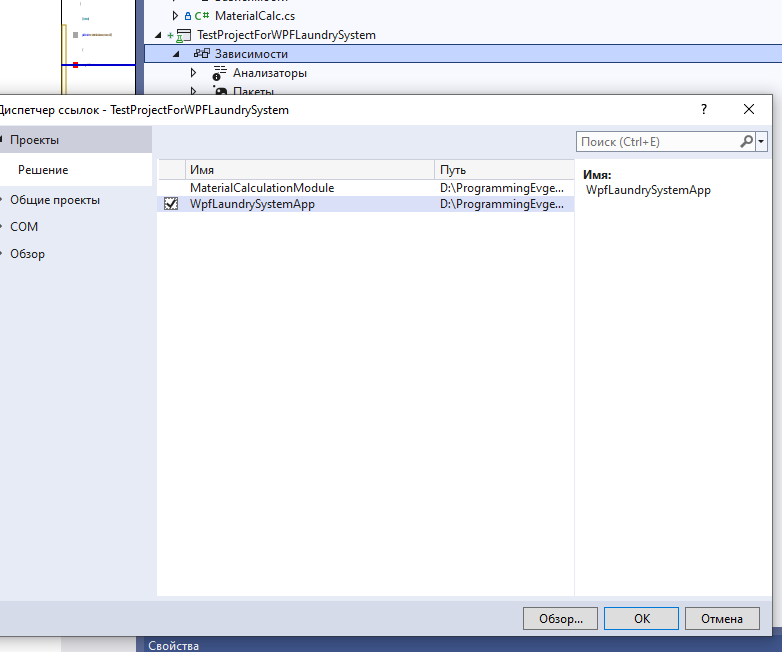
**Отчёт по 5 заданию**

Добавляем в решение Тестовый проект MSTest (Майкрософт)

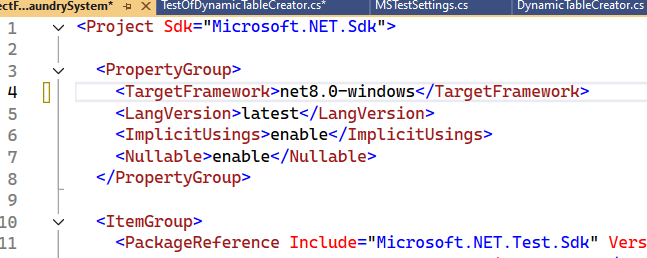




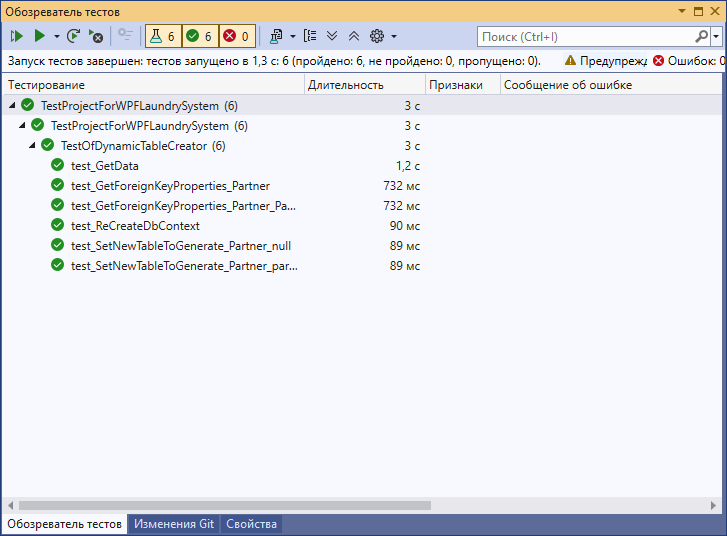
В зависимостях добавляем ссылку на проект



В настройках конфиге тестового проекта поименяем целевую платформу на net8.0-Уwindows



Пишем код тестов (код тестов в конце отчёта)  
После написания запускаем тесты.



**Тест-кейсы**

Тест 01 Открытие формы редактирования партнёра (OK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ПКП по партнёру | * Строка с партнёрам выделилась * Появилось контекстное меню с опциями: добавить, редактировать, удалить |
| 1. Нажать кнопку «Редактировать» | * Открывается окно редактирования * Данные из ранее выделенной строки партнёра введены в соответствующие поля нового окна |

Тест 02 Открытие формы добавления партнёра (OK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ПКП по партнёру | * Появилось контекстное меню с опциями: добавить, редактировать, удалить |
| 1. Нажать кнопку «Добавить» | * Открывается окно редактирования * Присутствуют пустые автоматически сгенерированные поля (все пустые, кроме идентификатора, он с значением 0) |

Тест 03 Открытие формы просмотра истории рейтинга партнёра (OK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ЛКМ по кнопке «Просмотреть» в столбце «Рейтинги» в строке с партнёром «ООО “ХимСервис Плюс”» | * Заголовок по центру сменился на «Рейтинги» * Строки с партнёрами пропали, и вместо них отображаются строки рейтингов * У строк рейтингов в столбце Партнёр указан «ООО “ХимСервис Плюс”» |

Тест 04 Попытка перейти назад до открытия других форм (NOK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ЛКМ по кнопке с символом «←» | * Кнопка подсветилась при наведении * После нажатия ничего не произошло, строки и заголовок окна не сменились |

Тест 05 Попытка удалить объект, на который ссылаются другие объекты (NOK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ПКП по строке с партнёром «ООО “ХимСервис Плюс”» | * Строка с партнёром выделилась * Появилось контекстное меню с опциями: добавить, редактировать, удалить |
| 1. Нажать кнопку «удалить» | * Выводиться сообщение об ошибке удаления * В сообщении сказано: Произошла ошибка удаления. Перед удаление объекта убедитесь, что у него нет зависимых (В таблице зависимые объекты можно посмотреть по кнопке "Просмотреть"), |

Тест 06 Попытка добавить объект, у которого не выбраны внешние ключи (NOK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ПКП по партнёру | * Появилось контекстное меню с опциями: добавить, редактировать, удалить |
| 1. Нажать кнопку «Добавить» | * Открывается окно редактирования * Присутствуют пустые автоматически сгенерированные поля (все пустые, кроме идентификатора, он с значением 0) |
| 1. Нажать кнопку «Сохранить» | * Выведется сообщение:   Ошибка при сохранен в поле “Вид партнёра” поле указано некорректно |

**Код** **тестов**

using System.Reflection;

using Microsoft.EntityFrameworkCore;

using WpfLaundrySystemApp.DBContext;

using WpfLaundrySystemApp.Models;

using WpfLaundrySystemApp.Modules;

namespace TestProjectForWPFLaundrySystem

{

[TestClass]

public sealed class TestOfDynamicTableCreator

{

/// <summary>

/// Тестирую, чтоб таблица корректно меняла тип генерируемой таблицы

/// на примере таблицы Partner без дополнительных условий where

/// </summary>

[TestMethod]

public void test\_SetNewTableToGenerate\_Partner\_null()

{

using (DynamicTableCreator tableCreator = new DynamicTableCreator(new LaundryDbContext()))

{

Type typeToChange = typeof(Partner);

object objectToGetWhereArgsFrom = null;

tableCreator.SetNewTableToGenerate(typeToChange, objectToGetWhereArgsFrom);

Assert.IsTrue(

tableCreator.TypeOfTheDynamicallyCreatedTable == typeToChange

&&

(tableCreator.WhereArguments is null || tableCreator.WhereArguments.Count == 0)

);

}

}

/// <summary>

/// Тестирую, чтоб таблица корректно меняла тип генерируемой таблицы

/// на примере таблицы Partner с дополнительными условиями where

/// </summary>

[TestMethod]

public void test\_SetNewTableToGenerate\_Partner\_partnerID\_1()

{

using (DynamicTableCreator tableCreator = new DynamicTableCreator(new LaundryDbContext()))

{

Type typeToChange = typeof(Partner);

int partnerId = 1;

Dictionary<PropertyInfo, object> whereArguments = new Dictionary<PropertyInfo, object>(1);

PropertyInfo propertToSet = typeToChange.GetProperty("PartnerId");

whereArguments[propertToSet] = partnerId;

tableCreator.SetNewTableToGenerate(typeToChange, whereArguments);

Assert.IsTrue(

tableCreator.TypeOfTheDynamicallyCreatedTable == typeToChange

&&

(tableCreator.WhereArguments is not null &&

tableCreator.WhereArguments.Count > 0 &&

tableCreator.WhereArguments.ContainsKey(propertToSet) &&

(((int)tableCreator.WhereArguments[propertToSet]) == partnerId)

)

);

}

}

/// <summary>

/// Негативный тест-кейс

/// Проверяю, что происходит исключение при попытке обратиться к disposed контексту, который пересоздаётся через ReCreateDbContext()

/// </summary>

[TestMethod]

public void test\_ReCreateDbContext()

{

LaundryDbContext initialLaundryDbContext = new LaundryDbContext();

using (DynamicTableCreator tableCreator = new DynamicTableCreator(initialLaundryDbContext))

{

tableCreator.ReCreateDbContext();

Assert.ThrowsException<ObjectDisposedException>(() => initialLaundryDbContext.Partners.ToList());

}

}

/// <summary>

/// Проверяю, что Получаются необходимые данные из dbContext

/// </summary>

[TestMethod]

public void test\_GetData()

{

LaundryDbContext laundryDbContext = new LaundryDbContext();

using (DynamicTableCreator tableCreator = new DynamicTableCreator(laundryDbContext))

{

Type typeOfPartner = typeof(Partner);

tableCreator.TypeOfTheDynamicallyCreatedTable = typeOfPartner;

List<Partner> partnersFromGetData = tableCreator.GetData().Select(p => p as Partner).ToList();

List<Partner> partnersFromDbContext = laundryDbContext.Partners.ToList();

CollectionAssert.AreEquivalent(expected: partnersFromDbContext, actual: partnersFromGetData);

}

}

/// <summary>

/// Проверяю, что Получаются необходимые внешние ключи из Partner, ссылающиеся на PartnerType

/// </summary>

[TestMethod]

public void test\_GetForeignKeyProperties\_Partner\_PartnerType()

{

using (DynamicTableCreator tableCreator = new DynamicTableCreator(new LaundryDbContext()))

{

Type typeThatReferences = typeof(Partner);

Type typeThatIsBeingReferencedTo = typeof(PartnerType);

PropertyInfo[] FKpropertiesExpected =

typeThatReferences

.GetProperties()

.Where(p => p.Name == "PartnerTypeId")

.ToArray();

PropertyInfo[] FKpropertiesActual =

tableCreator.GetForeignKeyProperties(typeThatReferences, typeThatIsBeingReferencedTo);

CollectionAssert.AreEquivalent(expected: FKpropertiesExpected, actual: FKpropertiesActual);

}

}

/// <summary>

/// Проверяю, что Получаются необходимые первичные ключи из Partner, ссылающиеся на PartnerType

/// </summary>

[TestMethod]

public void test\_GetForeignKeyProperties\_Partner()

{

using (DynamicTableCreator tableCreator = new DynamicTableCreator(new LaundryDbContext()))

{

Type typeThatHasPKs = typeof(Partner);

PropertyInfo[] PKpropertiesExpected =

typeThatHasPKs

.GetProperties()

.Where(p => p.Name == "PartnerId")

.ToArray();

PropertyInfo[] PKpropertiesActual =

tableCreator.GetPrimaryKeyProperties(typeThatHasPKs);

CollectionAssert.AreEquivalent(expected: PKpropertiesExpected, actual: PKpropertiesActual);

}

}

}

}