**Тестирование программного продукта**

1. Таблица тест-кейсов

| **№** | **Название теста** | **Входные данные** | **Действия** | **Ожидаемый результат** | **Тип теста** |
| --- | --- | --- | --- | --- | --- |
| 1 | Добавление нового партнера | Имя: "ООО Вест", ИНН: "111111111111", Телефон: "[+79999999999](tel:+79999999999)", Адрес: "ул. Вест, д. 21", Директор: "Вестиков А.А.", Почта: [info10@vest.ru](mailto:info10@vest.ru), Рейтинг: «5» | Нажать кнопку 'Сохранить' | Партнер успешно добавлен, отображается в списке | Позитивный |
| 2 | Создание заказа с существующим продуктом | PartnerID=1, ProductID=2, Quantity=10 | Нажать кнопку 'Создать заявку' | Заявка успешно создана, сумма = Цена\*Количество со скидкой | Позитивный |
| 3 | Расчет скидки партнера | PartnerID=1, общие продажи = 2000 | Нажать кнопку 'Рассчитать скидку' | Скидка = 0%, отображается корректно | Позитивный |
| 4 | Добавление партнера без имени | Имя пустое, ИНН="1234567890" | Нажать 'Сохранить' | Появляется предупреждение, запись не сохраняется | Негативный |
| 5 | Создание заказа с отрицательным количеством | PartnerID=1, ProductID=2, Quantity=-5 | Нажать 'Создать заявки' | Появляется ошибка, заявка не создается | Негативный |
| 6 | Добавление партнера с некорректным email | Имя="ООО Тест", Email="test@@mail" | Нажать кнопку  'Сохранить' | Появляется предупреждение, запись не сохраняется | Негативный |

1. Юнит-тесты

using Microsoft.VisualStudio.TestTools.UnitTesting;

using MasterPolApp;

namespace MasterPolApp.Tests

{

[TestClass]

public class PartnerTests

{

[TestMethod]

public void AddNewPartner\_ValidData\_ShouldReturnTrue()

{

var form = new PartnerForm();

form.SetPartnerData(

"ООО Вест",

"111111111111",

"+79999999999",

"ул. Вест, д. 21",

"Вестиков А.А.",

"info10@vest.ru",

5

);

bool result = form.ValidateInput();

Assert.IsTrue(result, "Должен успешно валидироваться корректный партнер");

}

[TestMethod]

public void AddPartner\_EmptyName\_ShouldReturnFalse()

{

var form = new PartnerForm();

form.SetPartnerData(

"",

"1234567890",

"+79999999999",

"ул. Тест, д.1",

"Тестов Т.Т.",

"test@mail.com",

5

);

bool result = form.ValidateInput();

Assert.IsFalse(result, "Проверка на пустое имя должна вернуть false");

}

[TestMethod]

public void AddPartner\_InvalidEmail\_ShouldReturnFalse()

{

var form = new PartnerForm();

form.SetPartnerData(

"ООО Тест",

"1234567890",

"+79999999999",

"ул. Тест, д.1",

"Тестов Т.Т.",

"test@@mail",

5

);

bool result = form.ValidateInput();

Assert.IsFalse(result, "Некорректный email должен не проходить валидацию");

}

[TestMethod]

public void CreateOrder\_ValidQuantity\_ShouldReturnTrue()

{

var orderForm = new PartnerOrderForm(1, "ООО Вест");

int validQuantity = 10;

bool result = orderForm.ValidateQuantity(validQuantity);

Assert.IsTrue(result, "Количество > 0 должно быть валидным");

}

[TestMethod]

public void CreateOrder\_NegativeQuantity\_ShouldReturnFalse()

{

var orderForm = new PartnerOrderForm(1, "ООО Вест");

int invalidQuantity = -5;

bool result = orderForm.ValidateQuantity(invalidQuantity);

Assert.IsFalse(result, "Отрицательное количество не должно проходить валидацию");

}

[TestMethod]

public void CalculateDiscount\_PartnerSales2000\_ShouldReturn0()

{

decimal totalSales = 2000;

decimal discount = PartnerOrderForm.CalculateDiscountBySales(totalSales);

Assert.AreEqual(0, discount, "Продажи 2000 → скидка 0%");

}

[TestMethod]

public void CalculateDiscount\_PartnerSales60000\_ShouldReturn10()

{

decimal totalSales = 60000;

decimal discount = PartnerOrderForm.CalculateDiscountBySales(totalSales);

Assert.AreEqual(10, discount, "Продажи 60000 → скидка 10%");

}

[TestMethod]

public void CalculateDiscount\_PartnerSales400000\_ShouldReturn15()

{

decimal totalSales = 400000;

decimal discount = PartnerOrderForm.CalculateDiscountBySales(totalSales);

Assert.AreEqual(15, discount, "Продажи 400000 → скидка 15%");

}

}

}

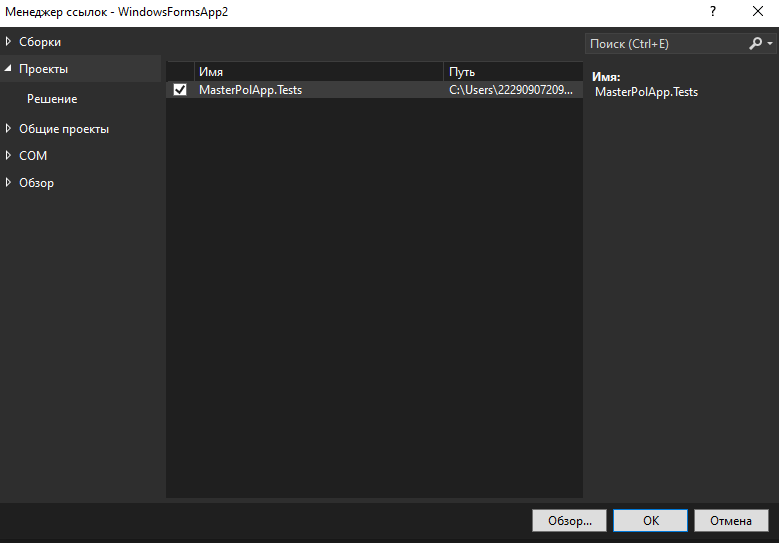


Рисунок 1 – Менеджер ссылок

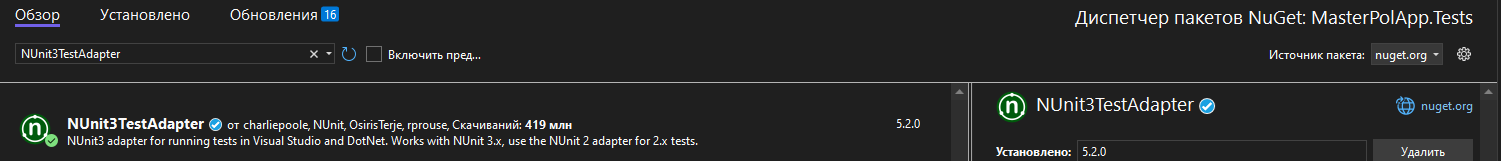


Рисунок 2 – Установка NUnit

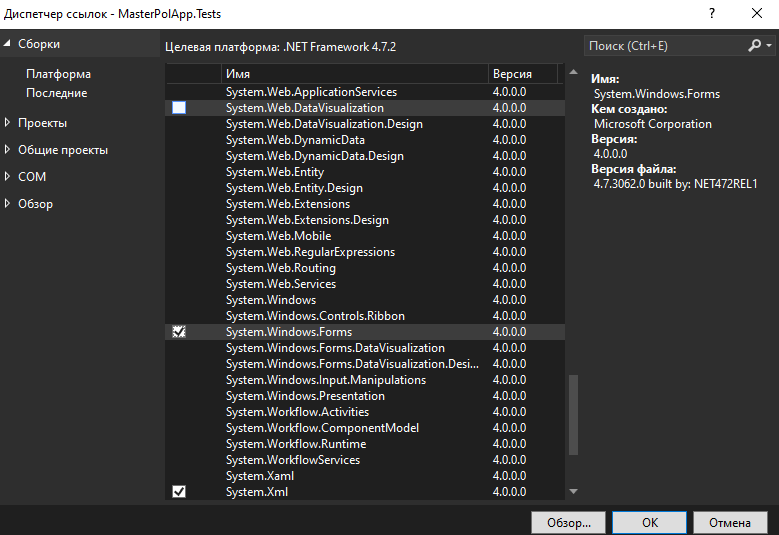


Рисунок 3 – Менеджер ссылок

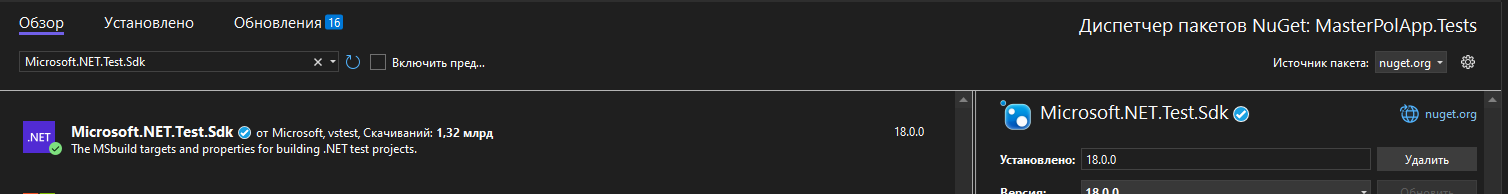


Рисунок 4 – Установка Test Sdk

1. Изображения программы

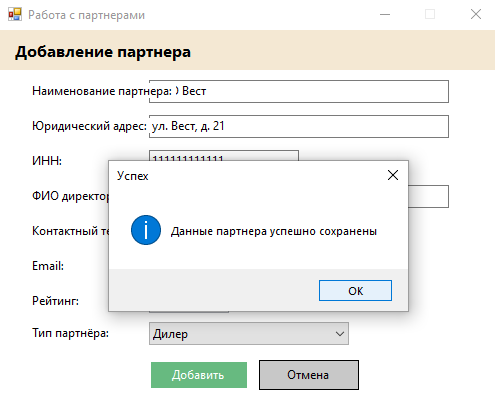


Рисунок 5 – Тест кейс 1

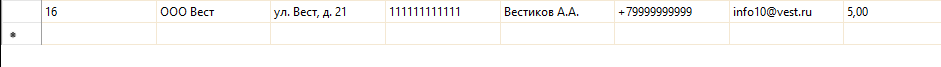


Рисунок 6 – Результат тест кейса 1

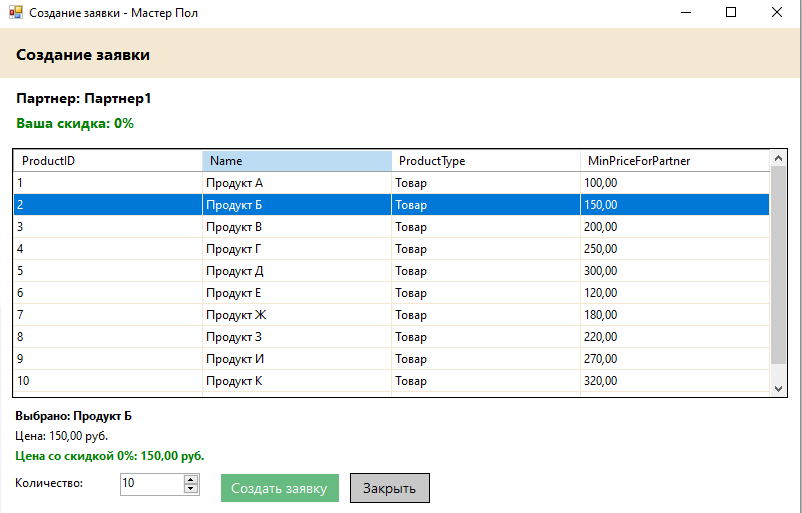


Рисунок 7 – Тест кейс 2

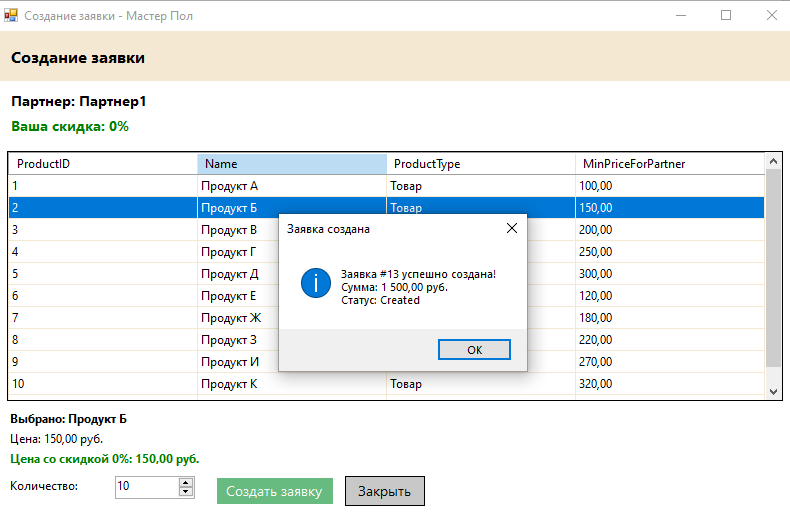


Рисунок 8 – Результат тест кейса 2

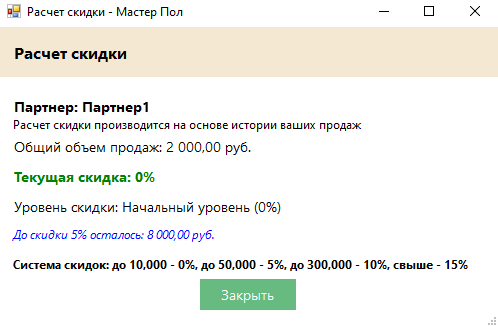


Рисунок 9 – Тест кейс 3

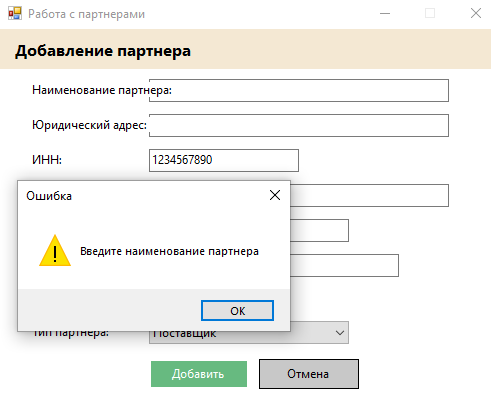


Рисунок 9 – Тест кейс 4

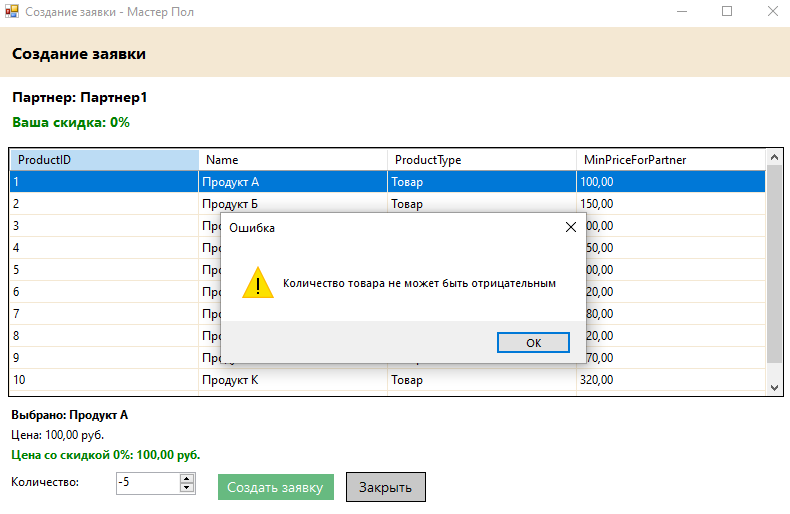


Рисунок 10 – Тест кейс 5

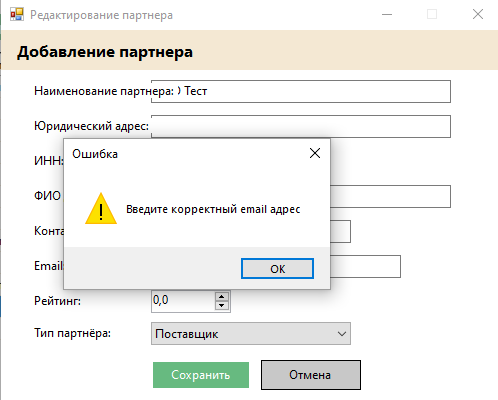
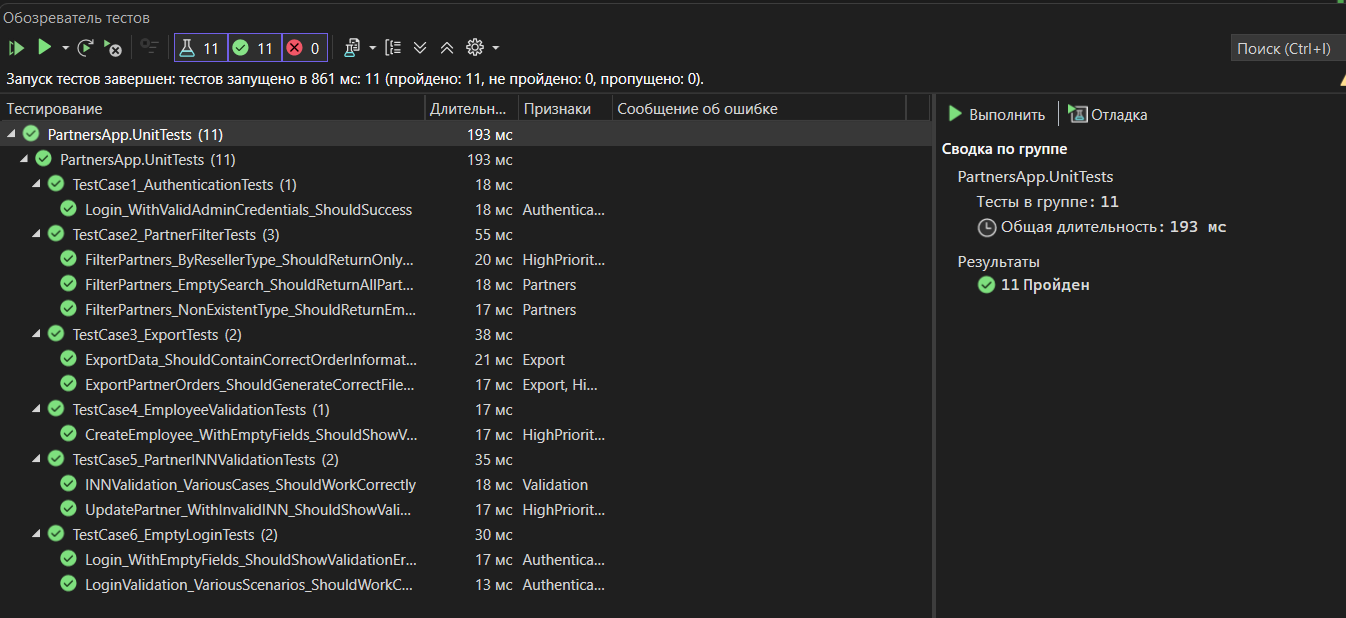


Рисунок 11 – Тест кейс 6

  
Рисунок 12 – Результат тестов