|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Министерство образования и науки Пермского края  государственное бюджетное профессиональное образовательное учреждение  «Пермский химико-технологический техникум»   |  |  | | --- | --- | |  | Специальность 09.02.07 «Информационные системы и программирование» |   **Отчет по учебной практике УП.02**  **ПМ.02 «**Осуществление интеграции программных модулей»    Сроки прохождения практики 2 декабря – 7 декабря 2024 года  Место прохождения практики ГБПОУ «ПХТТ»   |  |  | | --- | --- | |  | Выполнил студент гр. ИС-22-11 \_\_\_\_\_\_ /Р.В. Глухов/  группа подпись… расшифровка подписи  Руководитель практики  ГБПОУ «ПХТТ» \_\_\_\_\_\_\_\_\_\_\_\_ / М.А.Гилева /  подпись расшифровка подписи |   2023 |

**СОДЕРЖАНИЕ ОТЧЕТА ПО УЧЕБНОЙ ПРАКТИКЕ**

|  |  |
| --- | --- |
| Характеристика | **3** |
| Программа практики | **5** |
| 1. Отчет о выполненных работах | **6** |
| * 1. Задание 1 | **6** |
| * 1. Задание 2 | **21** |
| Список литературы | **30** |
| Дневник прохождения практики | **стр** |

# **ХАРАКТЕРИСТИКА**

Обучающийся (аяся) Глухов Руслан Владимирович Группы ИС-22-11

специальности СПО 09.02.07 «Информационные системы и программирование» проходил(а) учебную практику по профессиональному модулю **ПМ.01 «**Разработка модулей программного обеспечения для компьютерных систем» в период с «2» декабря 2024 г. по «7» декабря 2024 г на предприятии / в организации ГБПОУ «ПХТТ»

За время работы проявил(а) себя как ответственный/безответственный, исполнительный/неисполнительный, коммуникабельный/замкнутый, доброжелательный/наглый сотрудник.

В ходе выполнения всех видов работ обучающийся(аяся) показал(ла) сформированность следующих компетенций:

|  |  |
| --- | --- |
| **Наименование компетенции** | **Уровень (оптимальный/средний/ допустимый уровень)** |
| OK 0l. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. | оптимальный/средний/  допустимый / |
| ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности. | оптимальный/средний/  допустимый / |
| ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях. | оптимальный/средний/  допустимый / |
| ОК 04. Эффективно взаимодействовать и работать в коллективе и команде. | оптимальный/средний/  допустимый / |
| ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста. | оптимальный/средний/  допустимый / |
| ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения. | оптимальный/средний/  допустимый / |
| ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях. | оптимальный/средний/  допустимый / |
| ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. | оптимальный/средний/  допустимый / |
| ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках. | оптимальный/средний/  допустимый / |
| ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. | оптимальный/средний/  допустимый / |
| ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием. | оптимальный/средний/  допустимый / |
| ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств. | оптимальный/средний/  допустимый / |
| ПК 1.4. Выполнять тестирование программных модулей. | оптимальный/средний/  допустимый |
| ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода. | оптимальный/средний/  допустимый |

К работе относился\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Цели и задачи практики достигнуты / достигнуты не в полном объёме.

Руководитель практики от ГБПОУ «ПХТТ»

преподаватель Гилева Мария Александровна

|  |  |  |
| --- | --- | --- |
| Дата 7.12.2024 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | /М.А. Гилева/  Расшифровка подписи |

# **ПРОГРАММА УЧЕБНОЙ ПРАКТИКИ**

**по модулю ПМ.02 «**Осуществление интеграции программных модулей»

**по специальности 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование вида работ** | **Количество часов** |
|  | Инструктаж по технике безопасности и пожарной безопасности. Проектирование базы данных: разработка ER-диаграммы, оформление словаря данных. Проектирование функционала прикладного решения | 4 |
|  | Разработка прикладного программного обеспечения в соответствии с техническим заданием | 8 |
|  | Выполнение отладки программного решения, разработка сценариев для выполнения ручного тестирования. Разработка руководств. | 4 |
|  | Проектирование базы данных: разработка концептуальной модели. Разработка физической модели. Заполнение базы данных информацией | 4 |
|  | Разработка модулей программного обеспечения для компьютерных систем. Разработка алгоритмов. Отладка и тестирование. | 4 |
|  | Интеграция программного модуля. Осуществление автоматизированного и ручного тестирования. | 8 |
|  | Разработка руководств. | 2 |
|  | Оформление отчетной документации. | 2 |

Руководитель практики от ГБПОУ «ПХТТ»

преподаватель Гилева Мария Александровна

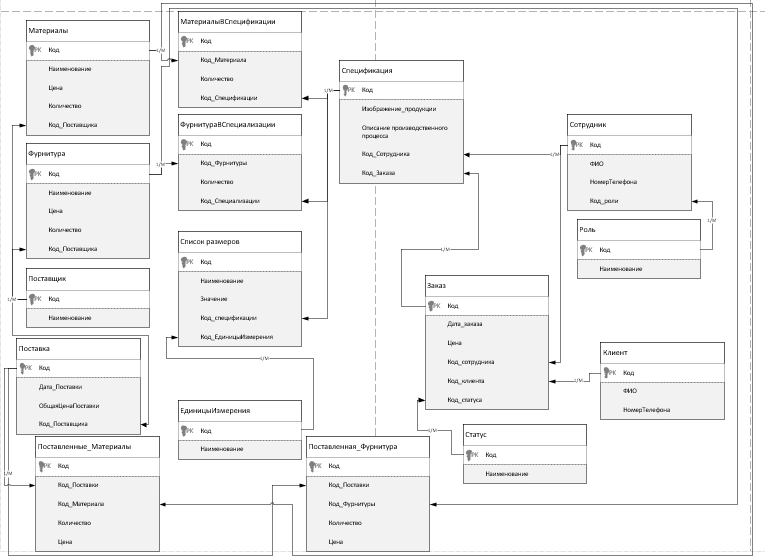
|  |  |  |
| --- | --- | --- |
| Дата 7.12.2024 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | /М.А. Гилева/  Расшифровка подписи |

# **ОТЧЕТ О ВЫПОЛНЕННЫХ РАБОТАХ**

**ЗАДАНИЕ 1**

**Отчет**

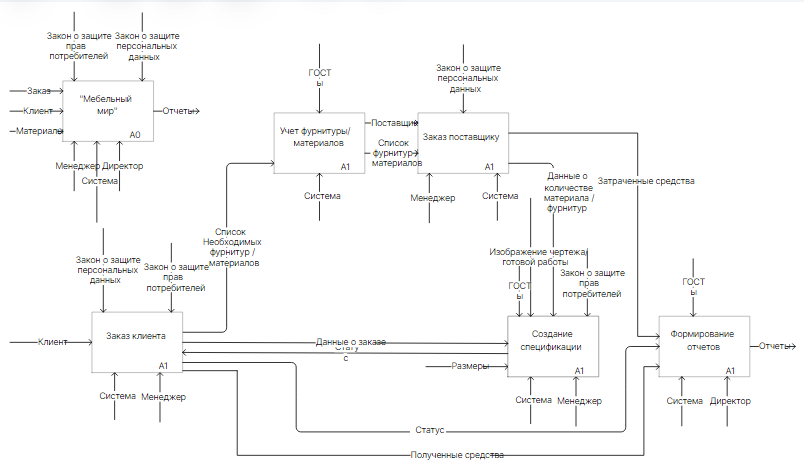
ER-диаграмма



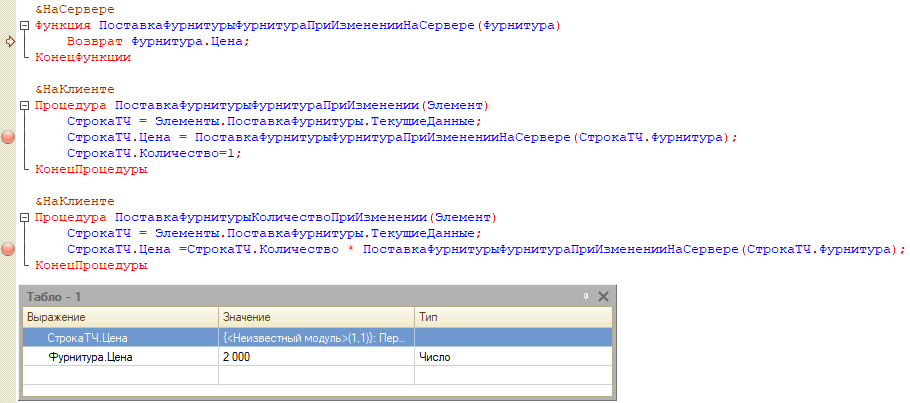
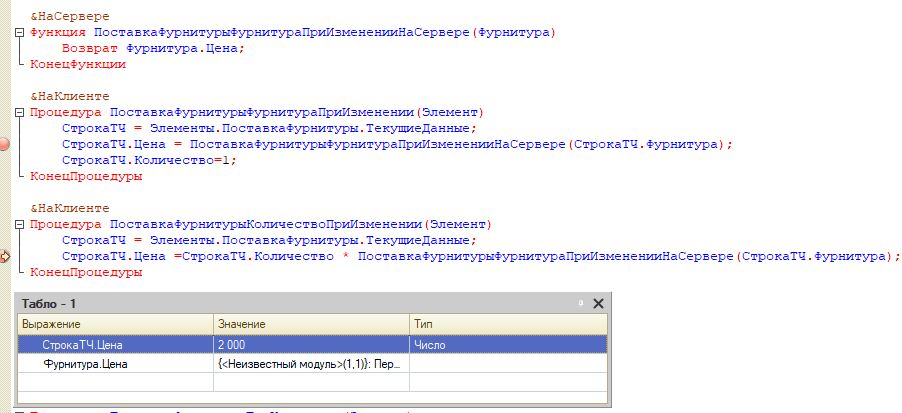
Словарь данных

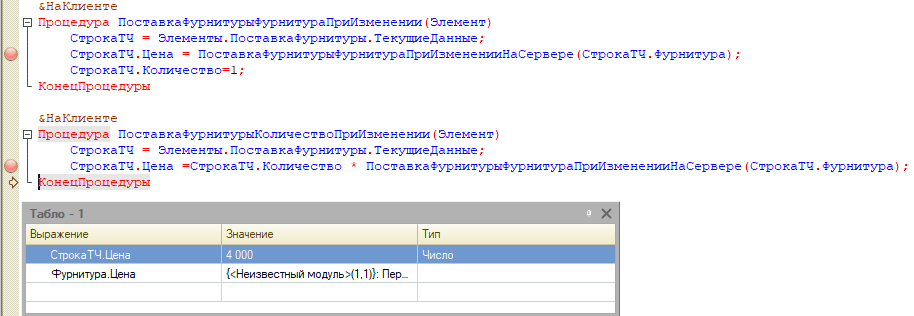


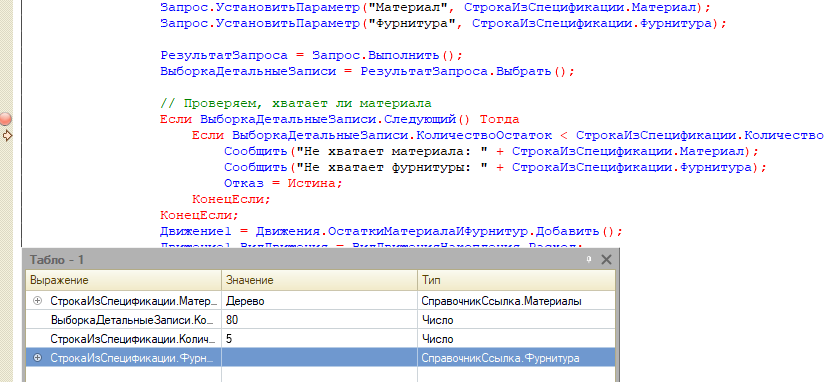
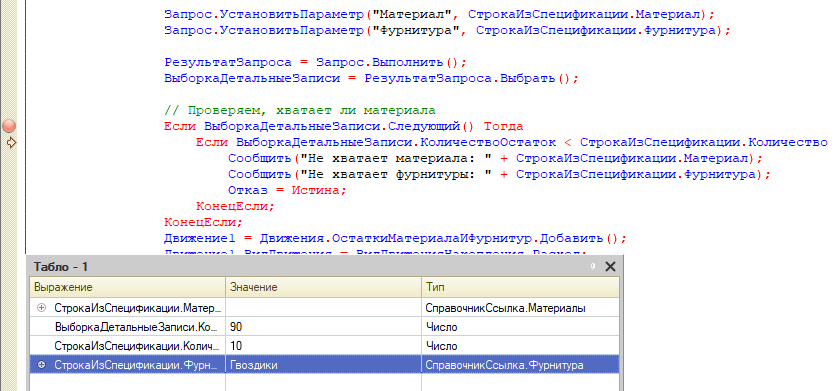
Функциональная диаграмма

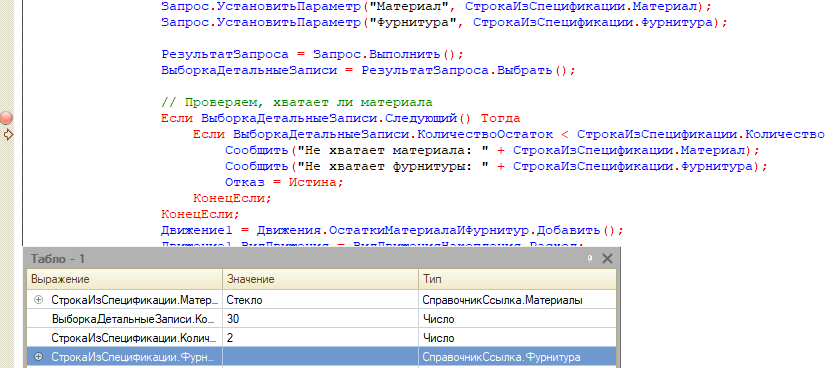
**

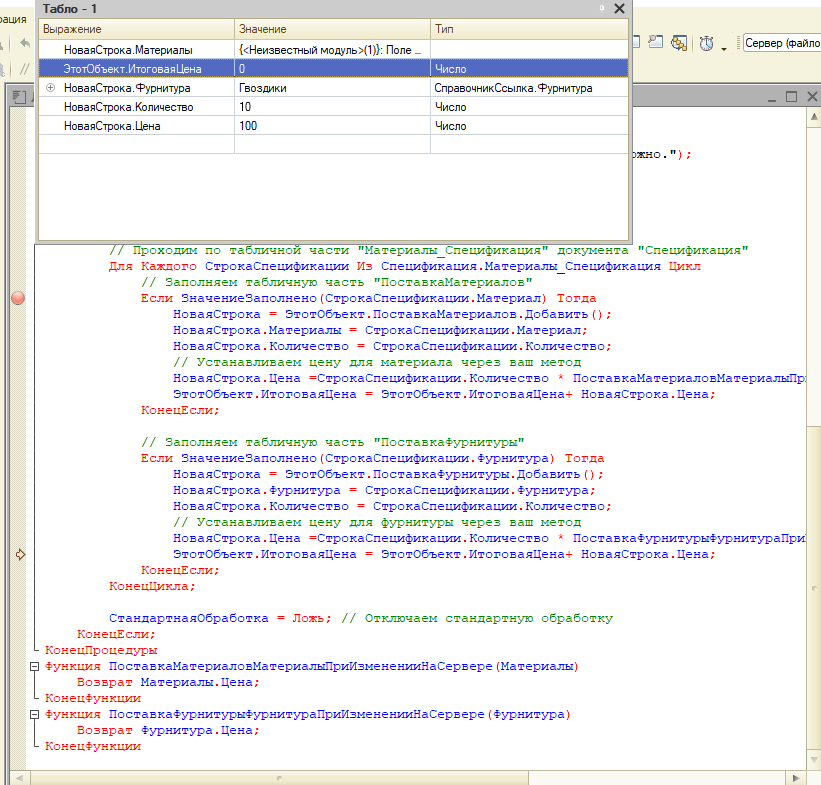
Выполнение отладки программного кода

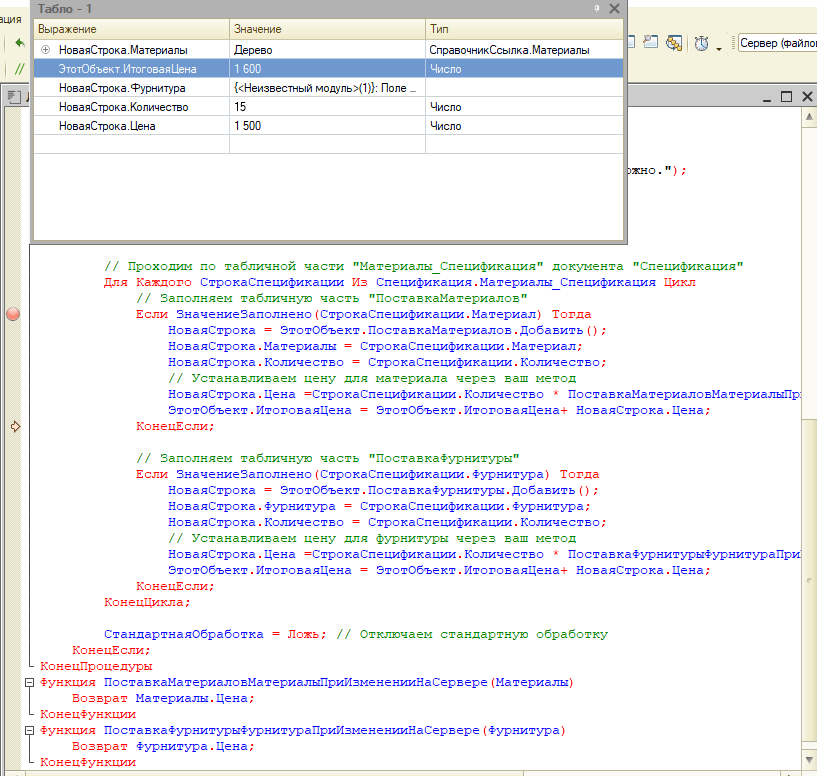


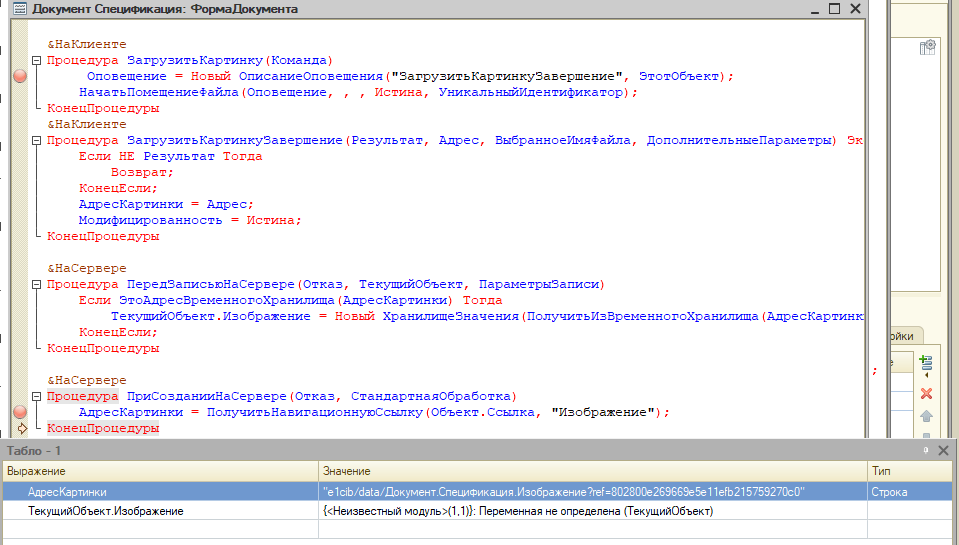


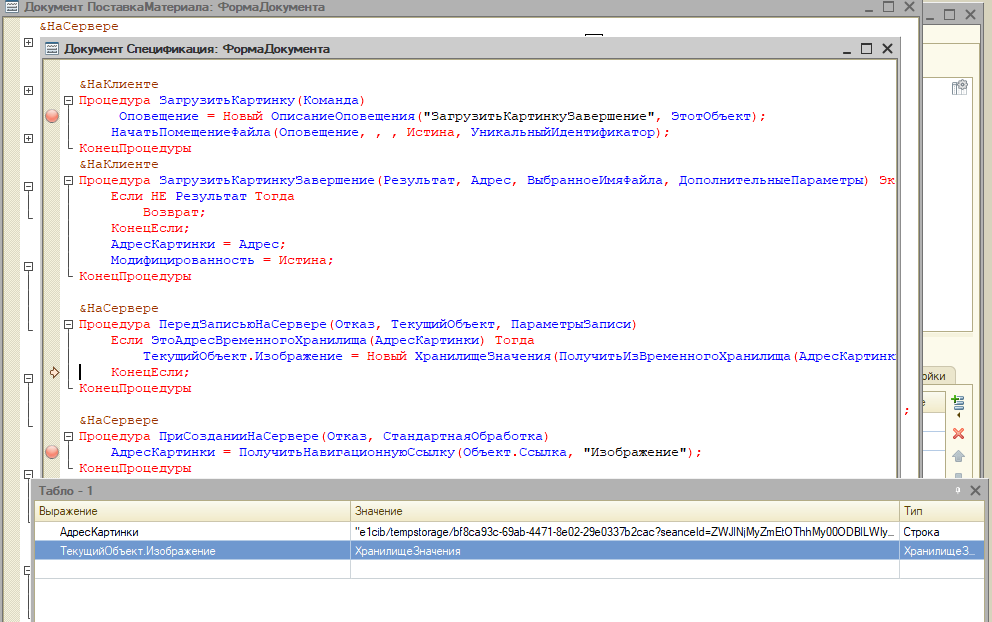


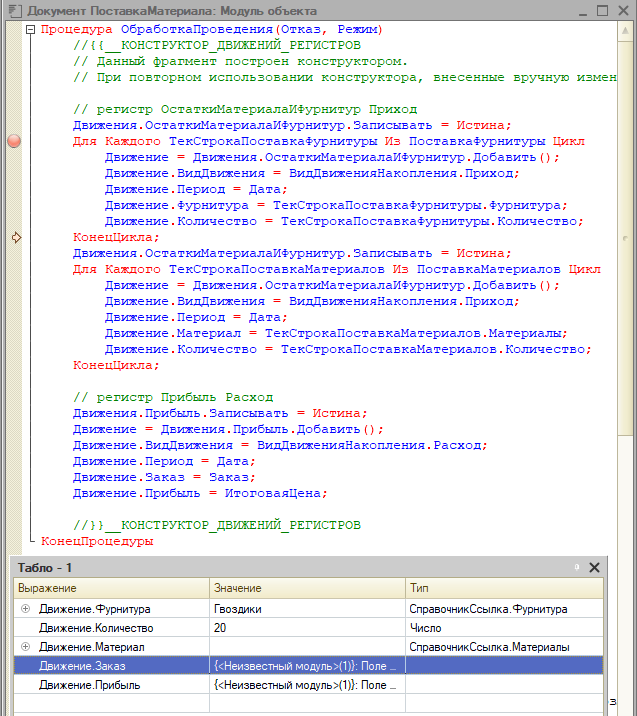


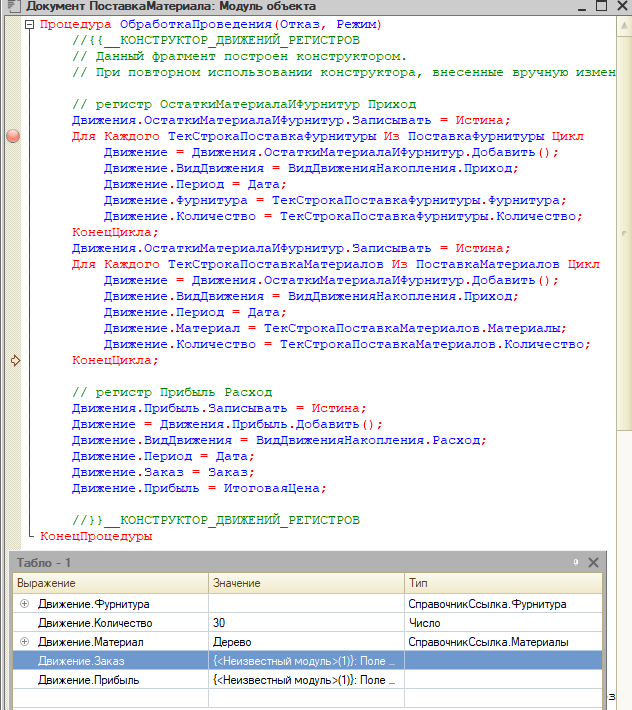


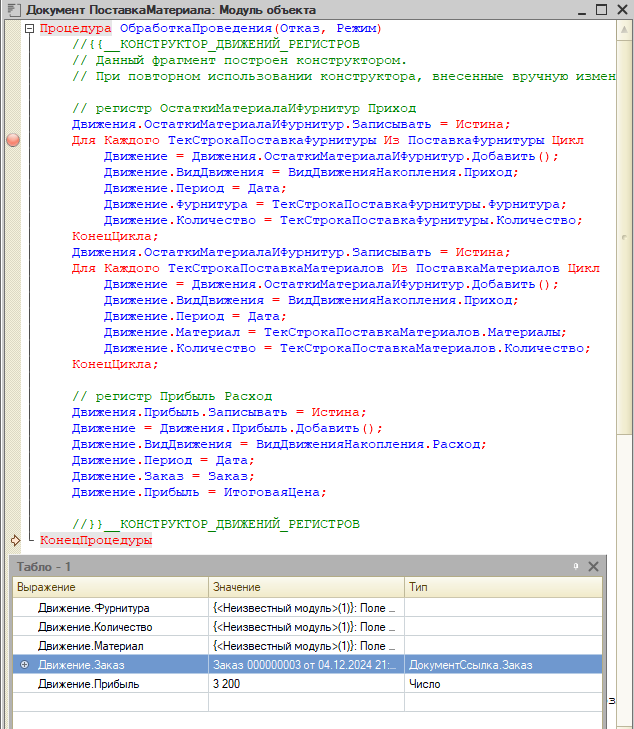




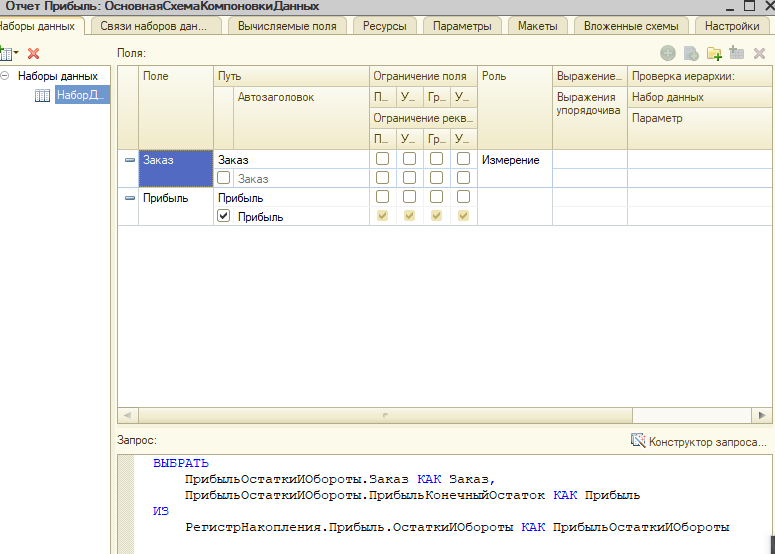


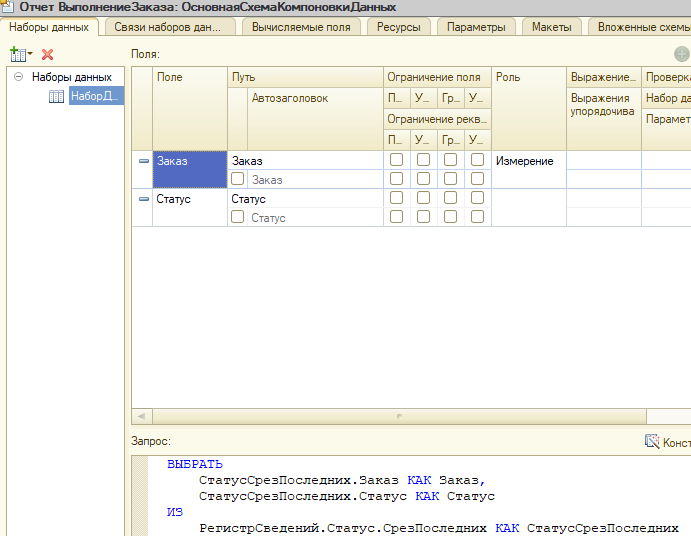




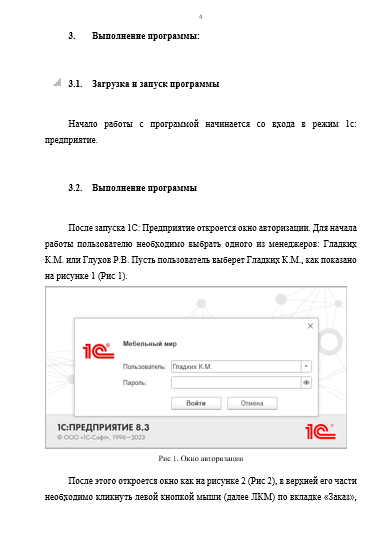
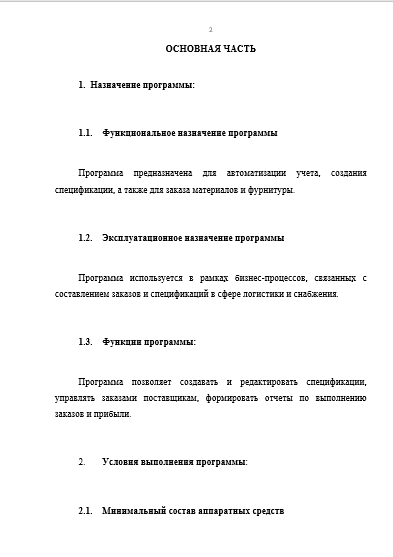


Программный код отчётов

**



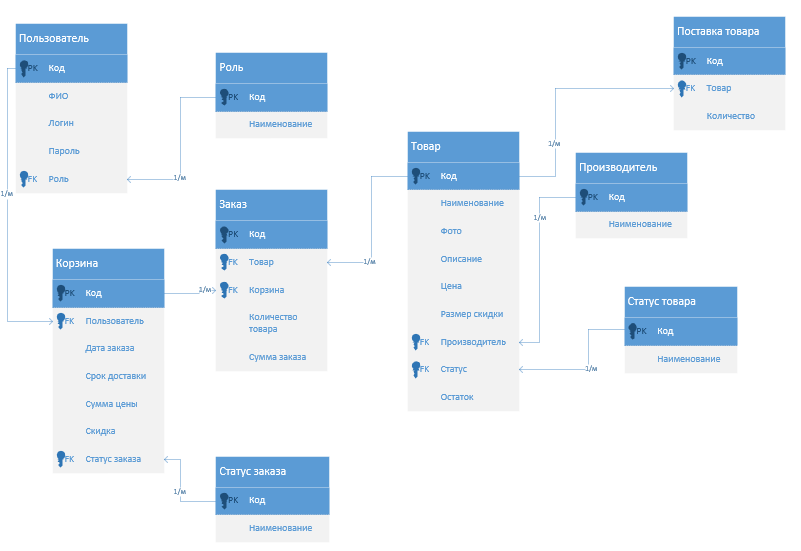
Руководство оператора



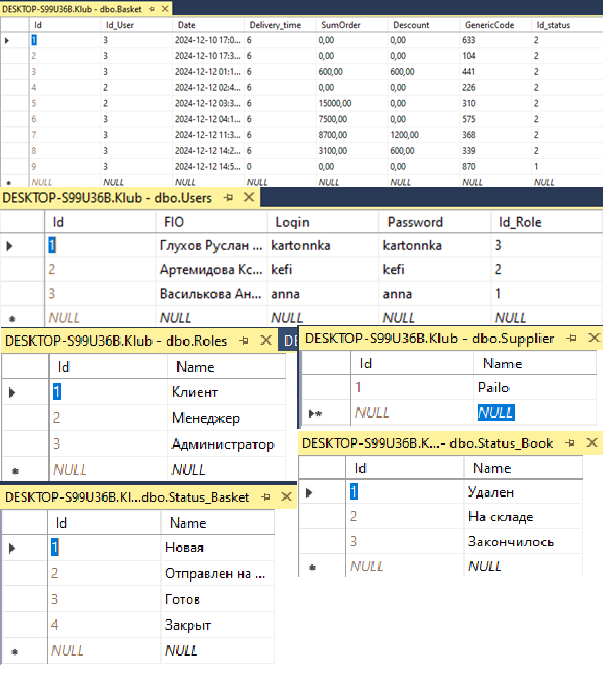
**ЗАДАНИЕ 2**

**Отчет**

ER-диаграмма

**

скриншот таблиц в СУБД

**

Программный код метода для подсчета скидки

ppublic decimal CalculateDiscountedPrice(decimal originalPrice, decimal? discount)

{

if (discount.HasValue && discount.Value > 0)

{

// Возвращаем цену с учетом скидки как decimal

return originalPrice \* (1 - discount.Value / 100);

}

else

{

// Если скидки нет, возвращаем исходную цену

return originalPrice;

}

}

Программный код добавления нового товара

namespace Klub.Windows.ManagerWindows

{

/// <summary>

/// Логика взаимодействия для AddBookWindow.xaml

/// </summary>

public partial class AddBookWindow : Window

{

BDEntities bd = new BDEntities();

public AddBookWindow()

{

InitializeComponent();

// Заполняем ComboBox производителями

autor.ItemsSource = bd.Suppliers.ToList();

autor.DisplayMemberPath = "Name"; // Замените на имя поля, которое вы хотите показывать в ComboBox

autor.SelectedValuePath = "ID"; // Это будет значение, которое будет использоваться в коде

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

string Name = nameTextBox.Text;

string Opis = descriptionTextBlock.Text;

// Проверка на пустые значения

if (string.IsNullOrEmpty(Name) || string.IsNullOrEmpty(Opis) || string.IsNullOrEmpty(priceTextBox.Text))

{

MessageBox.Show("Пожалуйста, заполните все поля.");

return;

}

decimal Prise;

if (!decimal.TryParse(priceTextBox.Text, out Prise))

{

MessageBox.Show("Пожалуйста, введите корректную цену.");

return;

}

// Получаем выбранного производителя из ComboBox

var selectedProizvoditel = (Supplier)autor.SelectedItem;

if (selectedProizvoditel == null)

{

MessageBox.Show("Пожалуйста, выберите производителя.");

return;

}

// Получаем путь к изображению

string imagePath = GetRelativeImagePath(image.Source != null ? image.Source.ToString() : string.Empty);

// Создаем новый товар

var newTovar = new Book

{

Name = Name,

Description = Opis,

Prise = Prise,

Id\_Status = 3, // предполагаем, что статус 3 означает "в наличии"

Remains = 0, // количество товара

Id\_Supplier = selectedProizvoditel.Id, // Устанавливаем производителя

Image = imagePath // Используем полученный путь к изображению

};

// Добавляем товар в базу данных

bd.Books.Add(newTovar);

bd.SaveChanges(); // Сохраняем изменения в базе данных

MessageBox.Show("Товар успешно добавлен.", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

this.Close(); // Закрываем окно после добавления товара

}

private string GetRelativeImagePath(string imageFileName)

{

// Убираем префикс "file:///" если он есть

if (imageFileName.StartsWith("file://"))

{

imageFileName = new Uri(imageFileName).LocalPath;

}

// Путь к папке с изображениями (относительно корня проекта)

string imagesFolderPath = System.IO.Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "Images");

// Проверка существования папки

if (!System.IO.Directory.Exists(imagesFolderPath))

{

// Если папка не существует, создаем её

System.IO.Directory.CreateDirectory(imagesFolderPath);

}

// Если файл изображения существует, сохраняем его в папке "Images"

if (System.IO.File.Exists(imageFileName))

{

string fileName = System.IO.Path.GetFileName(imageFileName);

string destinationPath = System.IO.Path.Combine(imagesFolderPath, fileName);

if (!System.IO.File.Exists(destinationPath))

{

System.IO.File.Copy(imageFileName, destinationPath);

}

// Возвращаем только имя файла с расширением

return fileName;

}

// Если файл не существует, возвращаем имя изображения по умолчанию

return "default\_image.png";

}

private void SelectImageButton\_Click(object sender, RoutedEventArgs e)

{

// Открываем диалог выбора изображения

Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog();

dlg.Filter = "Image Files|\*.jpg;\*.jpeg;\*.png;\*.bmp;\*.gif"; // фильтр для изображений

bool? result = dlg.ShowDialog();

if (result == true)

{

// Устанавливаем путь к изображению

string imagePath = dlg.FileName;

image.Source = new BitmapImage(new Uri(imagePath)); // Загружаем картинку в Image

}

}

}

}

Программный код unit-тестирования

namespace TestKlub

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void Test\_NormalCondition\_DiscountApplied()

{

// Arrange

MainWindow mainWindow = new MainWindow();

decimal originalPrice = 100m;

decimal? discount = 20m; // 20%

decimal expectedPrice = 80m; // 100 - 20%

// Act

var result = mainWindow.CalculateDiscountedPrice(originalPrice, discount);

// Assert

Assert.AreEqual(expectedPrice, result, "The discount calculation failed for normal conditions.");

}

[TestMethod]

public void Test\_ExtremeCondition\_NoDiscount()

{

// Arrange

MainWindow mainWindow = new MainWindow();

decimal originalPrice = 100m;

decimal? discount = 0m; // No discount

decimal expectedPrice = 100m;

// Act

var result = mainWindow.CalculateDiscountedPrice(originalPrice, discount);

// Assert

Assert.AreEqual(expectedPrice, result, "The discount calculation failed for an edge case of no discount.");

}

[TestMethod]

public void Test\_ExceptionalCondition\_NullDiscount()

{

// Arrange

MainWindow mainWindow = new MainWindow();

decimal originalPrice = 100m;

decimal? discount = null; // Null discount

decimal expectedPrice = 100m;

// Act

var result = mainWindow.CalculateDiscountedPrice(originalPrice, discount);

// Assert

Assert.AreEqual(expectedPrice, result, "The discount calculation failed for a null discount.");

}

}

}

Код генерации .pdf документа

private void GenerateOrderPdf(Basket basket)

{

string orderInfo = GetOrderInfo(basket); // Формируем информацию для QR-кода

// Генерация QR-кода

BarcodeWriter barcodeWriter = new BarcodeWriter

{

Format = BarcodeFormat.QR\_CODE,

Options = new ZXing.Common.EncodingOptions

{

Width = 300,

Height = 300,

Margin = 1,

Hints = { { ZXing.EncodeHintType.CHARACTER\_SET, "UTF-8" } }

}

};

var qrCodeImage = barcodeWriter.Write(orderInfo);

// Создаем PDF-документ

Document doc = new Document(PageSize.A4);

using (MemoryStream ms = new MemoryStream())

{

PdfWriter.GetInstance(doc, ms);

doc.Open();

// Добавляем QR-код в PDF

iTextSharp.text.Image qrImage = iTextSharp.text.Image.GetInstance(qrCodeImage, System.Drawing.Imaging.ImageFormat.Png);

qrImage.ScaleToFit(100f, 100f);

qrImage.SetAbsolutePosition(450f, 600f); // Устанавливаем позицию QR-кода на странице

doc.Add(qrImage);

doc.Close();

// Получаем путь к папке bin\Debug текущего проекта

string outputDirectory = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "Order\_" + basket.Id + ".pdf");

// Сохраняем PDF

File.WriteAllBytes(outputDirectory, ms.ToArray());

MessageBox.Show("PDF документ с QR-кодом был успешно создан! Можете найти его по адресу " + outputDirectory, "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

// Формируем строку с информацией о заказе для QR-кода

private string GetOrderInfo(Basket basket)

{

return $"Дата заказа: {DateTime.Now.ToShortDateString()}\n" +

$"Дата доставки: {DateTime.Now.AddDays(3).ToShortDateString()}\n" +

$"Номер заказа: {basket.Id}\n" +

$"В заказ входят следующие позиции: {string.Join(", ", basket.Orders.Select(z => z.Book.Name))}\n" +

$"Сумма заказа: {basket.SumOrder}\n" +

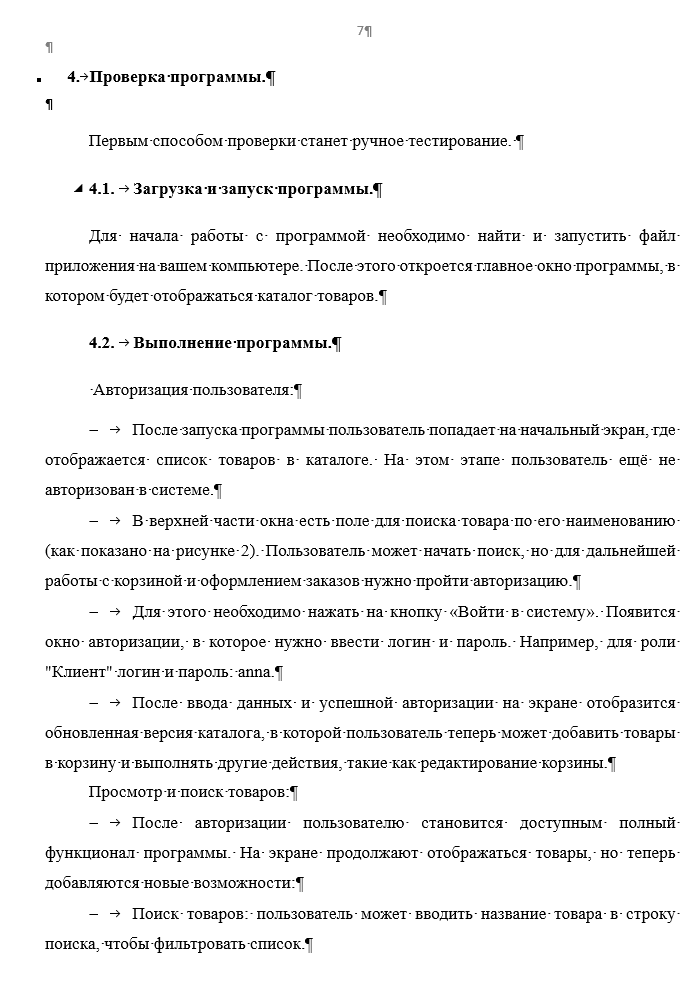
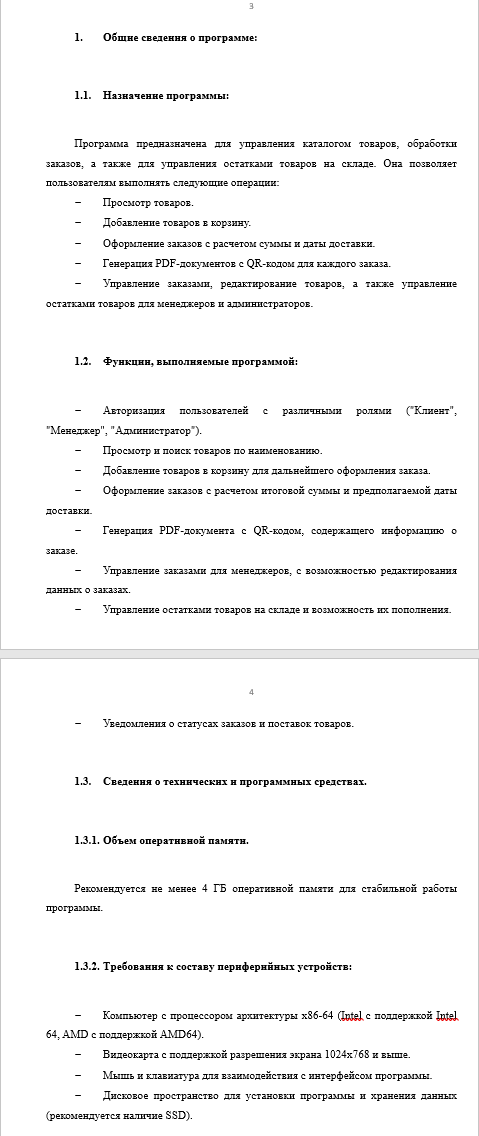
$"Скидка: {basket.Descount}\n" +

$"Генерируемый код доставки: {basket.GenericCode}";

}



Скриншоты руководства программиста



Github: <https://github.com/kerlondes/Klub.git> (Приложение магазин книг) и [https://github.com/kerlondes/TestKlub.git](https://github.com/kerlondes/TestKlub.git%20)  (unit тесты для приложения магазин книг)

**СПИСОК ЛИТЕРАТУРЫ**

1. Документация Microsoft по WPF (Windows Presentation Foundation) <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/>
2. Работа с базами данных в .NET <https://learn.microsoft.com/en-us/ef/>
3. Работа с QR-кодами в C# <https://github.com/micjahn/ZXing.Net>
4. Тестирование приложений <https://learn.microsoft.com/ru-ru/dotnet/core/testing/>