МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики  
Кафедра информационных технологий

Отчёт по индивидуальному заданию

**по дисциплине «Разработка приложений в интегрированных средах»**

Вариант №2

Работу выполнил  
студентка 35/2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Е. Баранова

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Проверил  
доцент кафедры, канд. техн. наук, доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Полупанов

Краснодар 2024

**Цель работы:** разработать клиентское приложение на языке С# для работы с базой данных формата PostgreSQL. Приложение должно иметь возможность добавлять, изменять, удалять данные из БД. По каждому отчёту необходимо отобразить диаграммы подходящего типа, согласно варианту задания.

**Задание**. Предприятие реализует товары клиентам на условиях 50% предоплаты (поле с отметкой). В договорах указываются товары, их количество и сумма. Отгрузка товара по договору может производиться частично. При отгрузке формируется товарная накладная (экспорт в MS Excel). За период для выбранных клиентов сформировать суммарный отчет согласно суммам товаров подлежащих отгрузке (есть предоплата) и отгруженных. За период для выбранных товаров сформировать отчёт, отображающий соотношение количества законтрактованных и оплаченных товаров (есть предоплата).

**Практическая часть.**

**1 Введение:**

Цель работы: Целью данной работы является разработка приложения на языке C# с использованием базы данных PostgreSQL для учета продаж и отгрузок товаров с возможностью анализа данных.

Задачи работы:

* Разработка структуры базы данных.
* Разработка пользовательского интерфейса.
* Реализация функций добавления, редактирования и удаления данных.
* Создание отчетов для анализа данных.
* Экспорт данных в формат Excel.

**2 Описание предметной области**

Предприятие, для которого разрабатывается приложение, реализует товары клиентам на условиях 50% предоплаты. Процесс реализации товаров включает следующие этапы:

Оформление договора: Клиент выбирает необходимые товары, их количество и оформляет договор с предприятием.

Внесение предоплаты: Клиент вносит 50% предоплаты за заказанные товары.

Отгрузка товара: После получения предоплаты предприятие отгружает товары клиенту. Отгрузка может производиться как полностью, так и частично.

Оплата оставшейся суммы: После получения товара клиент оплачивает оставшуюся сумму.

Формирование товарной накладной: При отгрузке товара формируется товарная накладная, которая является документом, подтверждающим факт передачи товара клиенту.

**3 Разработка структуры базы данных**

Для хранения данных о клиентах, товарах, заказах, предоплате и отгрузках была разработана реляционная база данных с использованием PostgreSQL.

3.1 Описание таблиц базы данных

База данных состоит из следующих таблиц:

Client: Хранит информацию о клиентах.

client\_id (integer): Уникальный идентификатор клиента (первичный ключ).

client\_name (text): Наименование клиента.

client\_address (text): Адрес клиента.

client\_phone (text): Телефон клиента.

Product: Хранит информацию о товарах.

product\_id (integer): Уникальный идентификатор товара (первичный ключ).

product\_name (text): Наименование товара.

product\_price (integer): Цена товара.

Orders: Хранит информацию о заказах.

order\_id (integer): Уникальный идентификатор заказа (первичный ключ).

order\_date (date): Дата оформления заказа.

client\_id (integer): Идентификатор клиента, оформившего заказ (внешний ключ).

total\_amount (integer): Общая сумма заказа.

OrderInfo: Хранит информацию о товарах в каждом заказе.

orderinfo\_id (integer): Уникальный идентификатор записи о товаре в заказе (первичный ключ).

order\_id (integer): Идентификатор заказа (внешний ключ).

product\_id (integer): Идентификатор товара (внешний ключ).

quantity (integer): Количество товара в заказе.

delivered (boolean): Флаг, указывающий, был ли товар отгружен.

payment\_50\_received (boolean): Флаг, указывающий, была ли получена предоплата в размере 50%.

Последовательности: В базе данных используются последовательности для автоматической генерации первичных ключей в таблицах (рисунок 1).

3.2 Диаграмма ER-модели

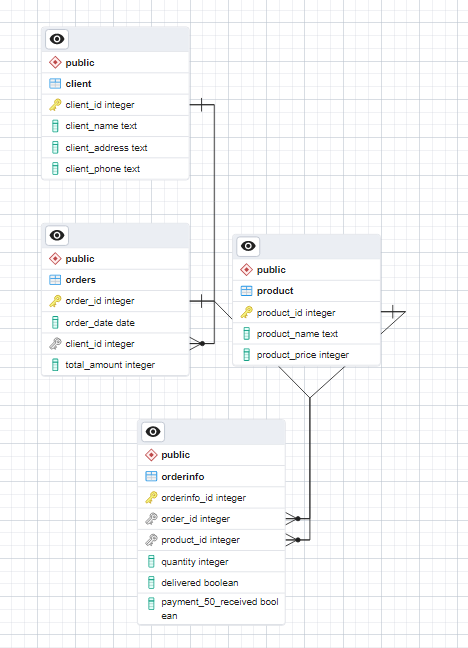


Рисунок 1 – ER-диаграмма

**4 Разработка приложения**

Для работы с базой данных было разработано приложение на языке C# с использованием библиотеки Npgsql для подключения к PostgreSQL.

4.1 Описание архитектуры приложения

Приложение представляет собой набор форм, каждая из которых предназначена для работы с определенным типом данных или для формирования отчетов. Главное окно приложения (Вариант 2) предоставляет доступ ко всем остальным формам (рисунок 2).

4.2 Главное окно приложения

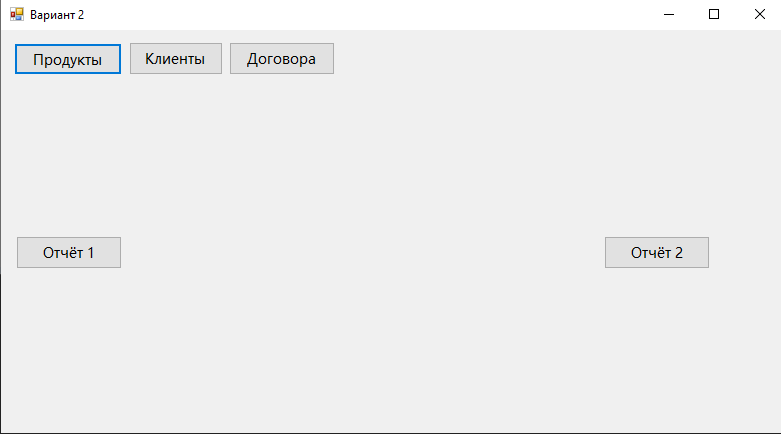


Рисунок 2 – Главная форма приложения

Главное окно приложения содержит кнопки для открытия следующих форм:

"Товары": Открывает форму FormProduct для работы с товарами.

"Клиенты": Открывает форму FormClient для работы с клиентами.

"Заказы": Открывает форму FormOrders для работы с заказами.

"Отчет1": Открывает форму FormProductReport для формирования отчета по товарам.

"Отчет2": Открывает форму FormClientOrderChart для формирования отчета по клиентам.

4.3 Описание форм

4.3.1 Форма для работы с клиентами (FormClient)

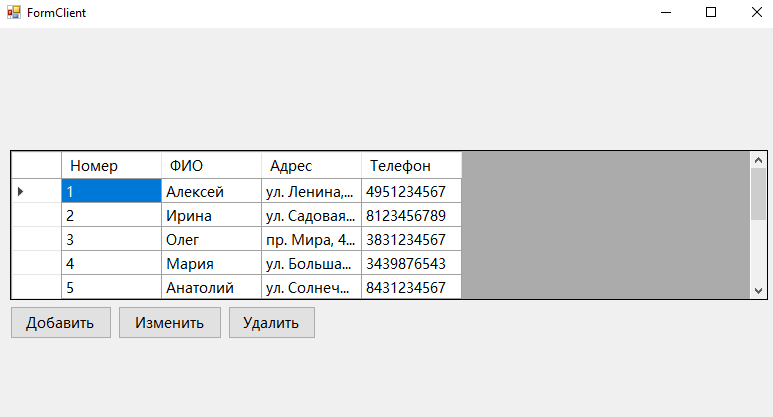


Рисунок 3 – Таблица клиентов

Форма FormClient предназначена для управления данными о клиентах. Она содержит следующие элементы управления:

dataGridViewClient: Таблица для отображения списка клиентов.

buttonClientShowAdd: Кнопка для добавления нового клиента.

buttonClientChange: Кнопка для редактирования выбранного клиента.

buttonClientDel: Кнопка для удаления выбранного клиента.

Пример кода обработчика кнопки "Добавить" (buttonClientShowAdd\_Click):

private void buttonClientShowAdd\_Click(object sender, EventArgs e)

{

FormClientAdd formClientAdd = new FormClientAdd(conn);

formClientAdd.ShowDialog();

Update();

}

Этот код создает экземпляр формы FormClientAdd для добавления нового клиента, отображает эту форму в модальном режиме (ShowDialog()), а после закрытия формы обновляет данные в таблице клиентов (Update()).

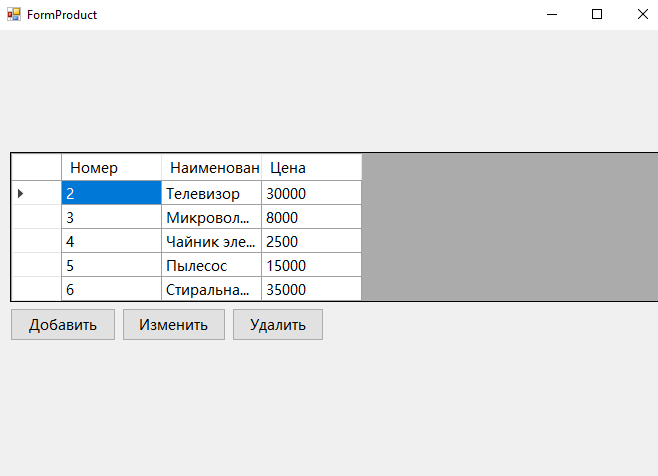
4.3.2 Форма для работы с товарами (FormProduct)

Рисунок 4 – Таблица всех продуктов

Форма FormProduct аналогична форме FormClient и предназначена для управления данными о товарах.

4.3.3 Форма для работы с заказами (FormOrders)

Форма FormOrders позволяет управлять данными о заказах, включая добавление новых заказов, редактирование существующих и удаление заказов. Также на этой форме доступна функция экспорта заказа в Excel.

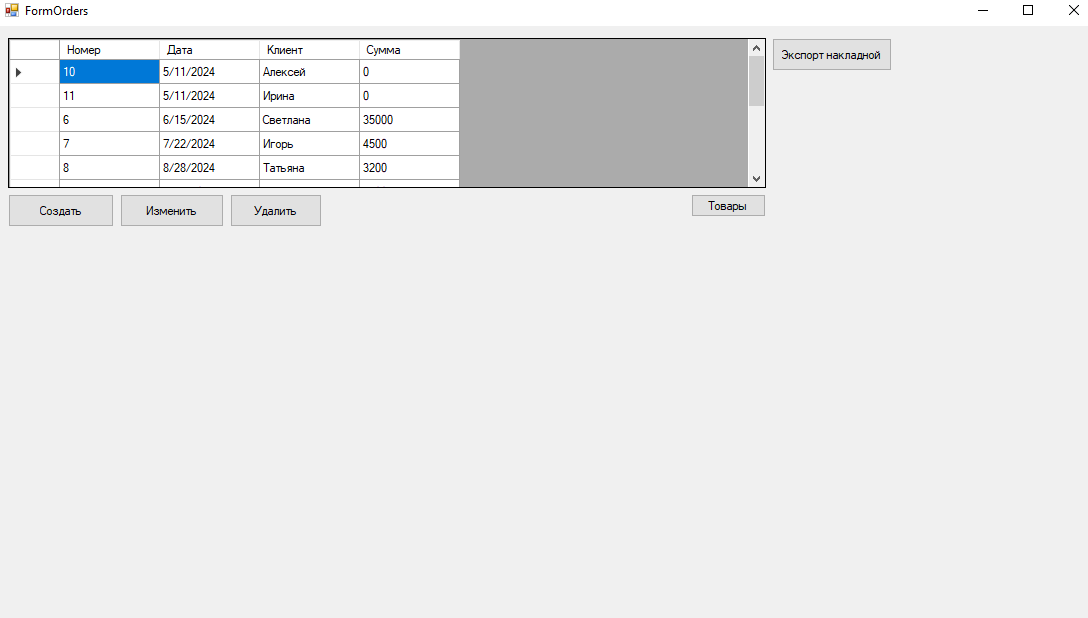


Рисунок 5 – Таблица всех заказов(договоров)

Функция экспорта заказа в Excel (ExportSelectedOrderToExcel)

public void ExportSelectedOrderToExcel(string filePath, int orderId)

{

// Код для получения данных о заказе и товарах в заказе из базы данных

using (var workbook = new XLWorkbook())

{

// Код для создания листа Excel, записи заголовков столбцов и данных о заказе

}

}

Эта функция формирует файл Excel с данными о выбранном заказе. Она получает данные о заказе и товарах из базы данных, создает новый лист Excel, записывает заголовки столбцов и данные о заказе в ячейки листа, а затем сохраняет файл Excel по указанному пути.

4.3.4 Форма для работы с товарами в заказе (FormOrderProduct)

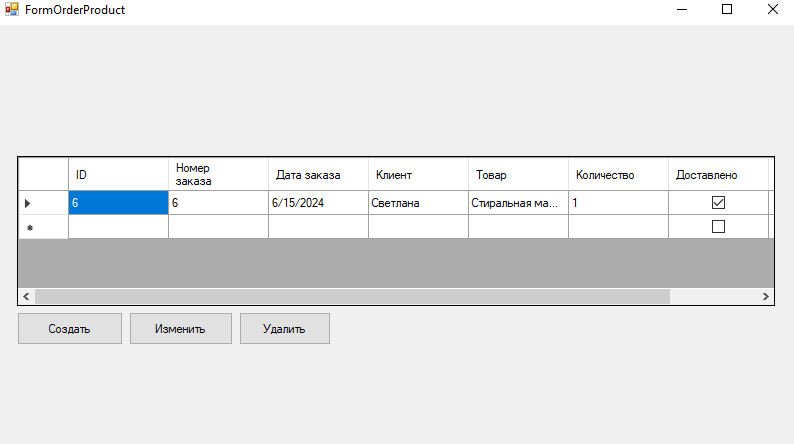


Рисунок 6 – Таблица заказа одного клиента

Форма FormOrderProduct позволяет добавлять товары в заказ, указывать их количество, отмечать товары как отгруженные, а также отмечать факт получения 50% предоплаты.

4.3.5 Форма для формирования отчёта по товарам (FormProductReport)

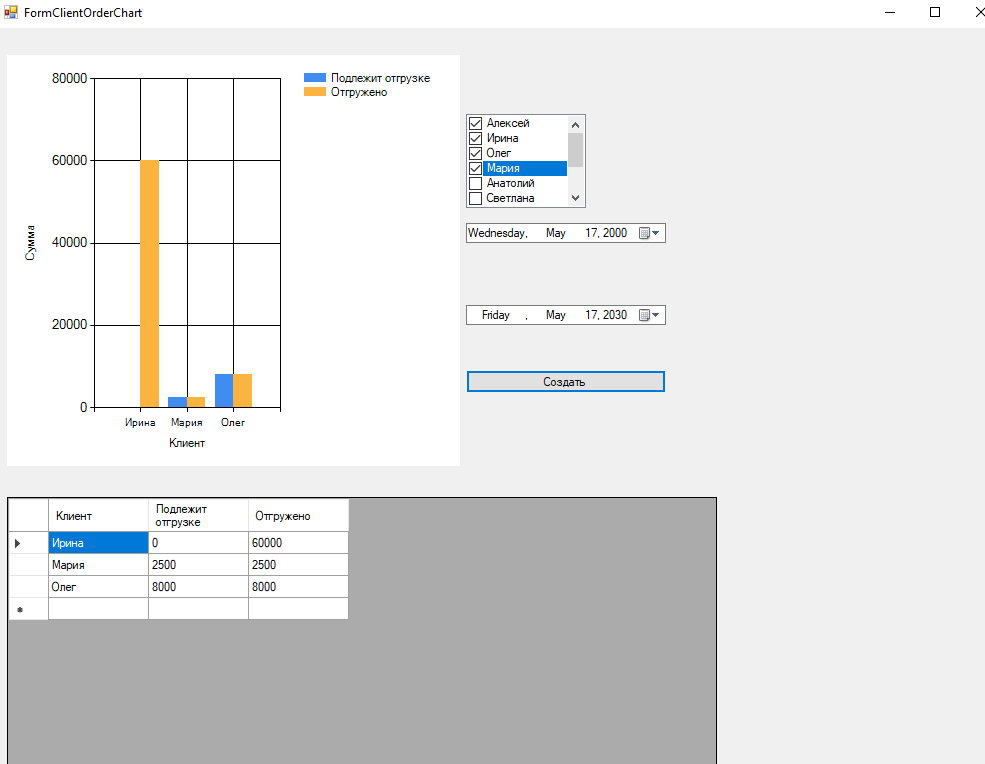


Рисунок 7 – Отчет по товарам

Форма FormProductReport позволяет сформировать отчет, отображающий соотношение количества законтрактованных и оплаченных товаров (с предоплатой) за выбранный период.

Пример кода обработчика кнопки "Сформировать отчёт" (buttonGenerateReport\_Click):

private void buttonGenerateReport\_Click(object sender, EventArgs e)

{

// Код для получения выбранных товаров и периода

string sql = $@"SELECT p.product\_name,

SUM(CASE WHEN o.order\_date BETWEEN @startDate AND @endDate THEN oi.quantity ELSE 0 END) AS total\_contracted,

SUM(CASE WHEN o.order\_date BETWEEN @startDate AND @endDate AND oi.payment\_50\_received THEN oi.quantity ELSE 0 END) AS total\_paid

FROM Product p

LEFT JOIN OrderInfo oi ON p.product\_id = oi.product\_id

LEFT JOIN Orders o ON oi.order\_id = o.order\_id

WHERE p.product\_name IN ({string.Join(",", selectedProducts.Select(c => $"'{c}'"))})

GROUP BY p.product\_name";

using (var cmd = new NpgsqlCommand(sql, conn))

{

// Код для выполнения запроса и заполнения DataGridView данными

}

}

Этот код формирует SQL-запрос для получения данных о законтрактованных и оплаченных товарах за выбранный период, выполняет запрос к базе данных и отображает полученные данные в таблице dataGridView1.

4.3.6 Форма для формирования отчёта по клиентам (FormClientOrderChart)

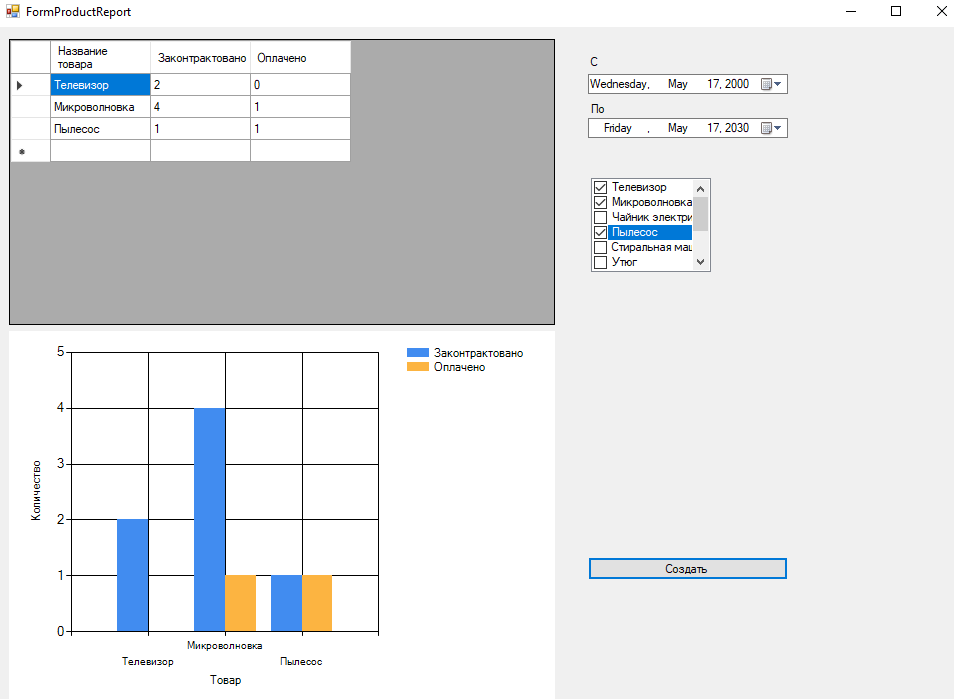


Рисунок 8 – Отчет по клиентам

Форма FormClientOrderChart позволяет сформировать суммарный отчет по клиентам за выбранный период, отображающий суммы товаров, подлежащих отгрузке (с предоплатой) и отгруженных.

**5 Подключение к базе данных**

Для подключения к базе данных PostgreSQL в приложении используется библиотека Npgsql. Библиотека Npgsql - это .NET-драйвер с открытым исходным кодом для PostgreSQL, который позволяет приложениям .NET подключаться к базам данных PostgreSQL и взаимодействовать с ними.

Создание подключения к базе данных:

conn = new NpgsqlConnection("Host=localhost;Username=postgres;Password=123;Database=nastya");

conn.Open();

5.1 Пример создания таблицы

CREATE TABLE IF NOT EXISTS public.orders

(

order\_id integer NOT NULL DEFAULT nextval('orders\_order\_id\_seq'),

order\_date date NOT NULL,

client\_id integer NOT NULL,

total\_amount integer NOT NULL,

CONSTRAINT orders\_pkey PRIMARY KEY (order\_id),

CONSTRAINT orders\_client\_id\_fkey FOREIGN KEY (client\_id)

REFERENCES public.client (client\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

);

5.2 Пример заполнения таблицы

INSERT INTO public.client (client\_id, client\_name, client\_address, client\_phone)

VALUES

(1, 'Алексей', 'ул. Ленина, 12, Москва', '4951234567'),

(2, 'Ирина', 'ул. Садовая, 30, Санкт-Петербург', '8123456789'),

(3, 'Олег', 'пр. Мира, 45, Новосибирск', '3831234567'),

(4, 'Мария', 'ул. Большая, 92, Екатеринбург', '3439876543'),

(5, 'Анатолий', 'ул. Солнечная, 7, Казань', '8431234567'),

(6, 'Светлана', 'пр. Ленина, 64, Нижний Новгород', '8317654321'),

(7, 'Игорь', 'ул. Заречная, 18, Челябинск', '3511234567'),

(8, 'Татьяна', 'пр. Победы, 56, Омск', '3812234567'),

(9, 'Наталья', 'ул. Луговая, 33, Самара', '8461234567'),

(10, 'Владимир', 'пр. Космонавтов, 90, Ростов-на-Дону', '8631234567');

**6 Пример работы программы**

6.1 Формирование отчёта по товарам

Для формирования отчета по товарам необходимо выполнить следующие действия:

Открыть форму FormProductReport.

Выбрать товары, по которым требуется сформировать отчет, установив флажки в списке товаров.

Выбрать период, за который требуется сформировать отчет, используя элементы управления dateTimePicker1 и dateTimePicker2.

Нажать кнопку "Сформировать отчет".

В результате будет сформирован отчет, отображающий соотношение количества законтрактованных и оплаченных товаров (с предоплатой) за выбранный период. Отчет представлен в виде таблицы и графика.

6.2 Формирование отчёта по клиентам

Для формирования отчета по клиентам необходимо выполнить следующие действия:

Открыть форму FormClientOrderChart.

Выбрать клиентов, по которым требуется сформировать отчет, установив флажки в списке клиентов.

Выбрать период, за который требуется сформировать отчет, используя элементы управления dateTimePicker1 и dateTimePicker2.

Нажать кнопку "Сформировать отчет".

В результате будет сформирован отчет, отображающий суммы товаров, подлежащих отгрузке (с предоплатой) и отгруженных за выбранный период. Отчет представлен в виде таблицы и графика.

Вывод: в рамках данной работы было разработано приложение, которое позволяет автоматизировать учет продаж и отгрузок товаров на предприятии. Приложение предоставляет возможность вести учет клиентов, товаров, заказов, предоплаты и отгрузок, а также формировать отчеты для анализа данных.

**Приложение**

https://github.com/BaranovaAnastasia17/RpvisIndiv