**Государственное бюджетное профессиональное образовательное учреждение**

**«Нижегородский Губернский колледж»**

Методическая комиссия «Информатика и вычислительная техника»

Допущен к защите:

преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_ Л.В. Мухина,

\_\_\_\_\_\_\_\_\_\_\_\_ Е.П. Голубева,

«21»\_\_марта\_\_2025 г.

**ОТЧЕТ** **ПО**

**УЧЕБНОЙ ПРАКТИКЕ**

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Руководитель: \_\_\_\_\_\_\_\_\_\_ Мухина Л.В., Голубева Е.П. 21.03.2025 г.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Колиниченко Е.С. 21.03.2025 г.

Специальность, группа: 09.02.07, 43П

Нижний Новгород

2025 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc192890896)

[РАЗРАБОТКА МОДУЛЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 4](#_Toc192890897)

[Модуль 1. Создание базы данных. 4](#_Toc192890898)

[Модуль 2. Разработка desktop-приложения. 6](#_Toc192890899)

[Модуль 3. Разработка библиотеки 17](#_Toc192890900)

[Модуль 4. Тестирование программных решений. 18](#_Toc192890901)

[ЗАКЛЮЧЕНИЕ 25](#_Toc192890902)

[Приложение А 26](#_Toc192890903)

[Приложение Б 38](#_Toc192890904)

# **ВВЕДЕНИЕ**

Основной целью практики являлось освоение общих и профессиональных компетенций, включая формирование алгоритмов разработки программных модулей, их реализацию в соответствии с техническим заданием, отладку с использованием специализированных инструментов, тестирование, рефакторинг и оптимизацию кода, а также разработку модулей для мобильных платформ.

Для достижения поставленной цели были поставлены следующие задачи:

1. Создание и заполнение базы данных в соответствии с техническим заданием.
2. Разработка desktop-приложения.
3. Разработка библиотеки в соответствии с техническим заданием.
4. Проведение отладки и тестирования готового приложения.

Итогом учебной практики стал отчет, содержащий описание выполненных заданий, а также графические материалы, такие как диаграммы, таблицы и иллюстрации.

# **РАЗРАБОТКА МОДУЛЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

## **Модуль 1. Создание базы данных.**

В рамках этого модуля были выполнены следующие задачи:

1. ER-диаграмма была разработана с использованием СУБД Postree SQL. Диаграмма включает таблицы, связи между ними, атрибуты и ключи. На данном этапе основное внимание уделялось структуре данных, а не их типам. Сама диаграмма представлена ниже.

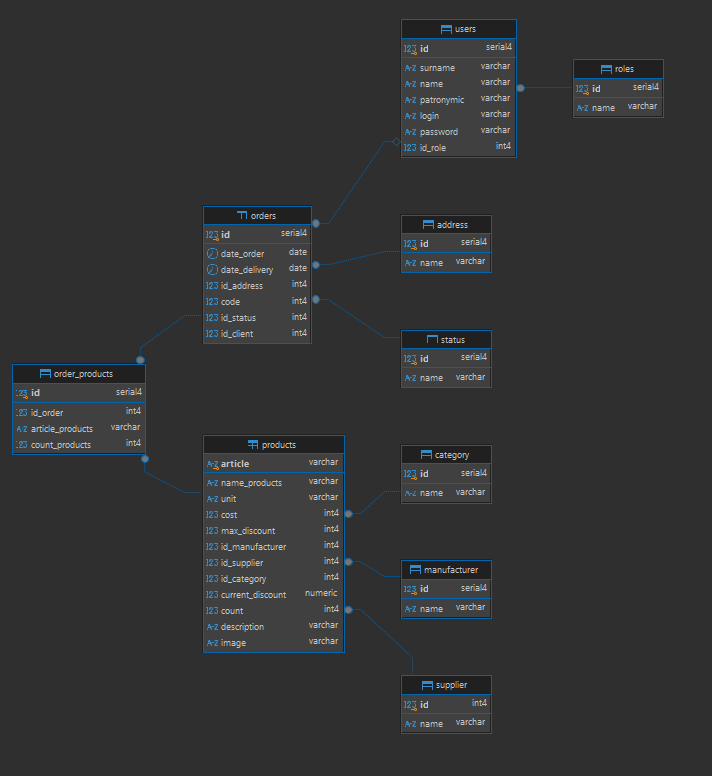


Рисунок № 1 - Разработанная ER-диаграмма для системы.

1. На основе описания предметной области была создана база данных в СУБД PostgreSQL. База данных была нормализована до третьей нормальной формы, что обеспечило ссылочную целостность данных. Скрипт созданной базы данных смотреть в [Приложении А](#_Приложение_А), листинг 1.
2. Импорт данных: Заказчик предоставил файлы с данными, помеченные как "import", для переноса в новую систему. Данные были подготовлены и успешно загружены в разработанную базу данных. Скрипт импорта данных смотреть в [Приложении А, листинг 2](#_Приложение_А).

## **Модуль 2. Разработка desktop-приложения.**

В рамках данного задания была выполнена разработка алгоритма подсистемы для работы магазина по продаже канцелярских товаров. Алгоритмы были реализованы в виде кода программного продукта с использованием среды разработки Visual Studio и языка программирования C#, с фреймворком Avalonia. Эта подсистема реализует в себе работу со следующими экранами:

1. Авторизация.

Реализовано окно входа, которое является первым экраном, видимым пользователю. Пользователь может ввести логин и пароль или перейти к просмотру товаров в роли гостя. После успешной авторизации в интерфейсе отображается ФИО пользователя. В случае неудачной попытки авторизации, система запрашивает ввод CAPTCHA, состоящей из 4 символов с графическим шумом. После второй неудачной попытки система блокирует вход на 10 секунд. Скрины предоставлены на рисунках № 2-4.



Рисунок 2. Форма авторизации.

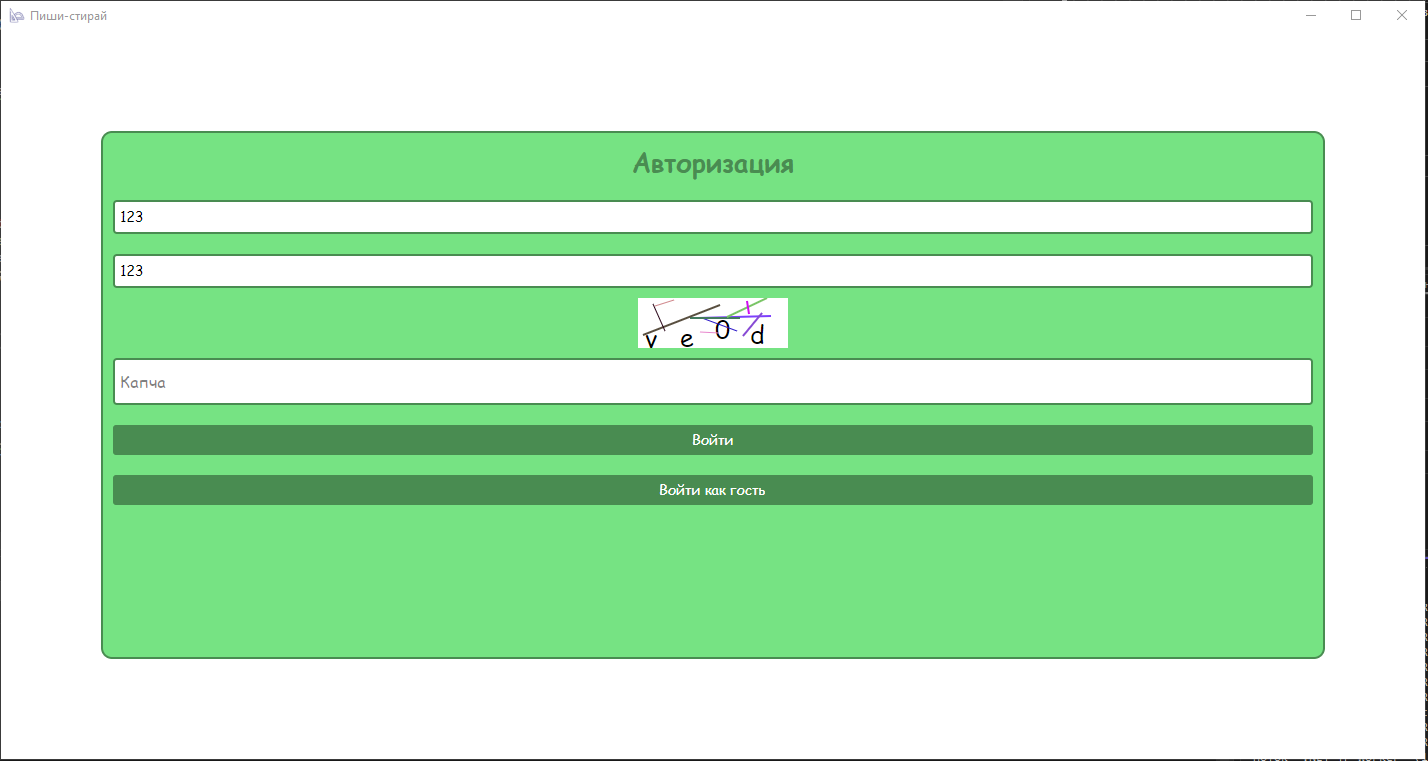


Рисунок 3. Попытка ввода не правильных логина и пароля. Вывод капчи.

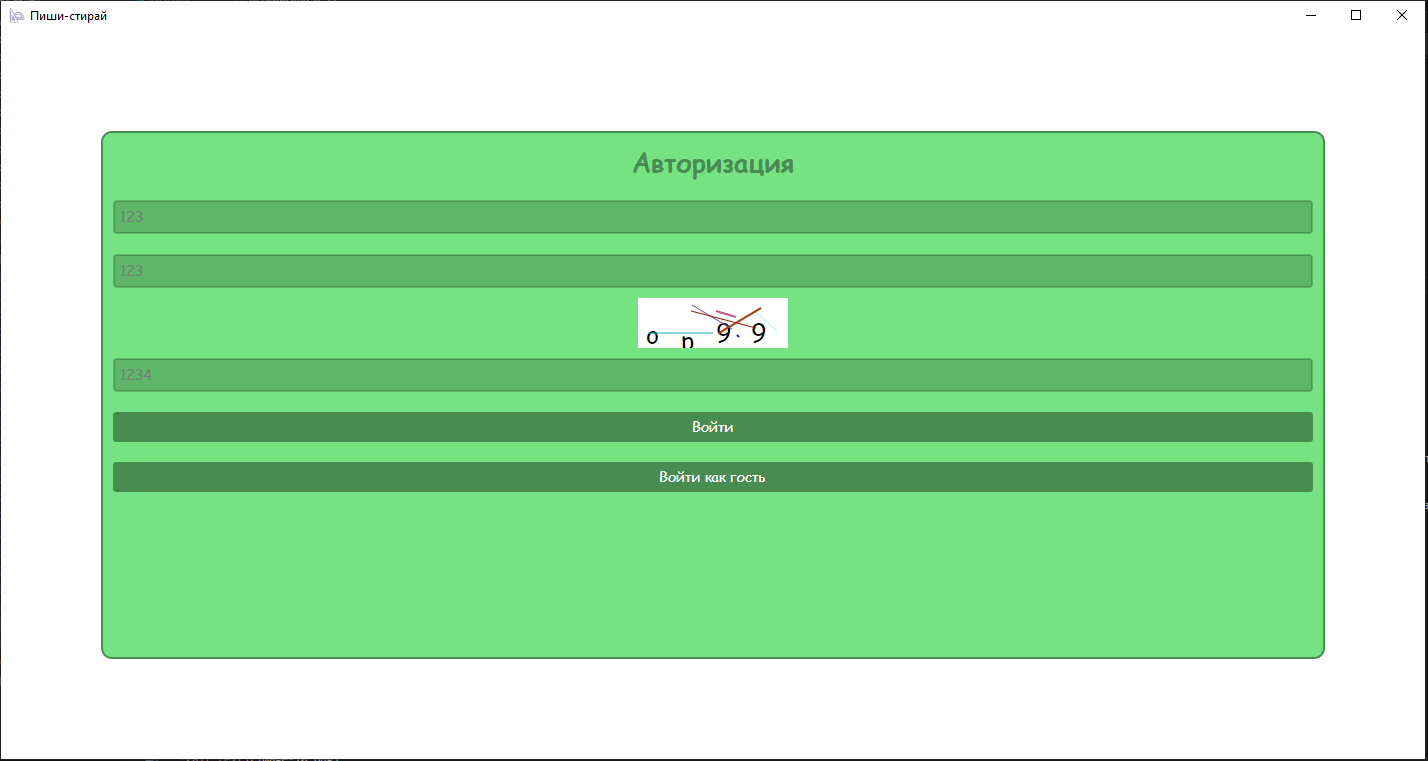


Рисунок 4. Блокировка полей на 10 секунд.

1. Список товаров.

Реализован вывод списка товаров с актуальной информацией об остатках на складах. При отсутствии изображения товара выводится картинка-заглушка. Строки с товарами подсвечиваются в зависимости от размера скидки. Реализованы функции сортировки по стоимости, фильтрации по размеру скидки и поиска по наименованию товара. Все изменения применяются в реальном времени без необходимости нажатия дополнительных кнопок. В верхней части окна отображается количество выведенных данных и общее количество записей.

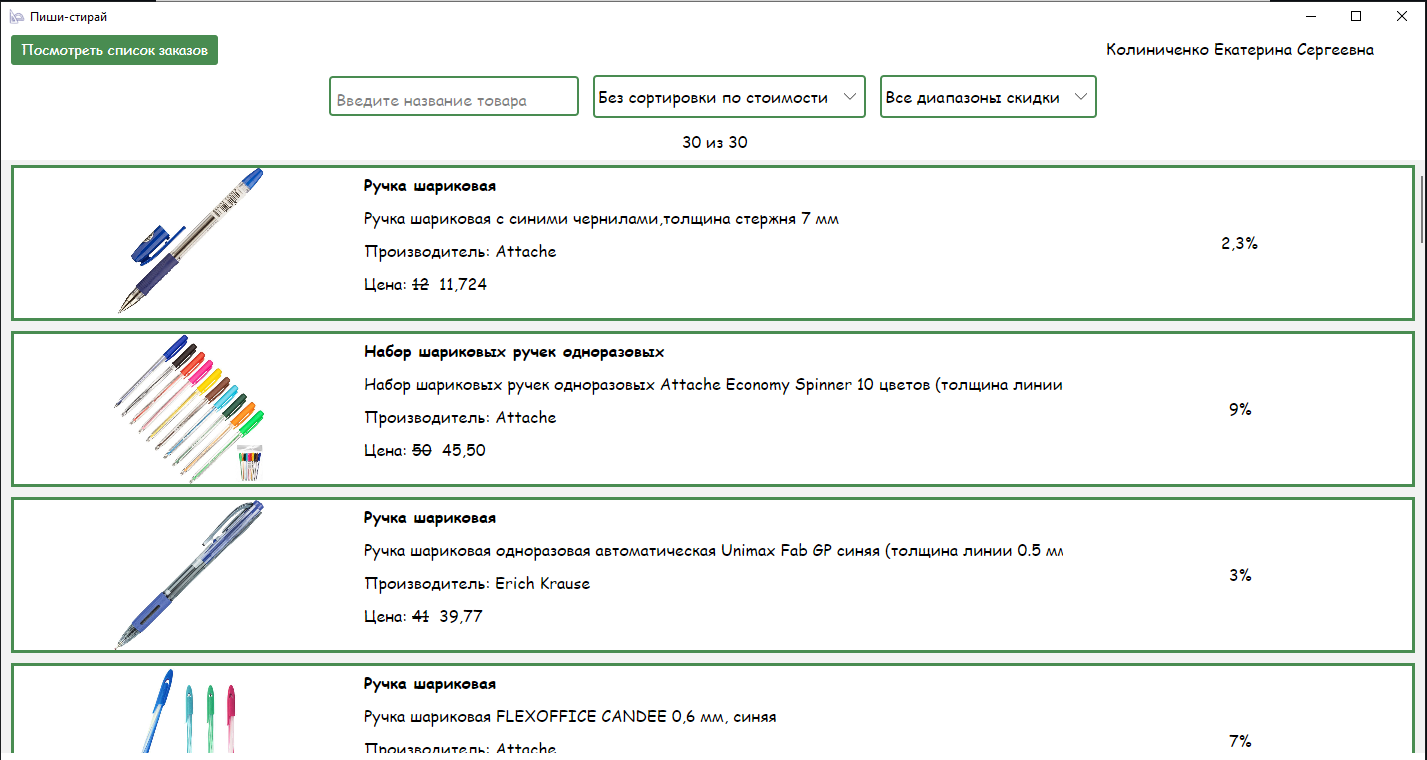


Рисунок 5. Страница авторизированного пользователя.

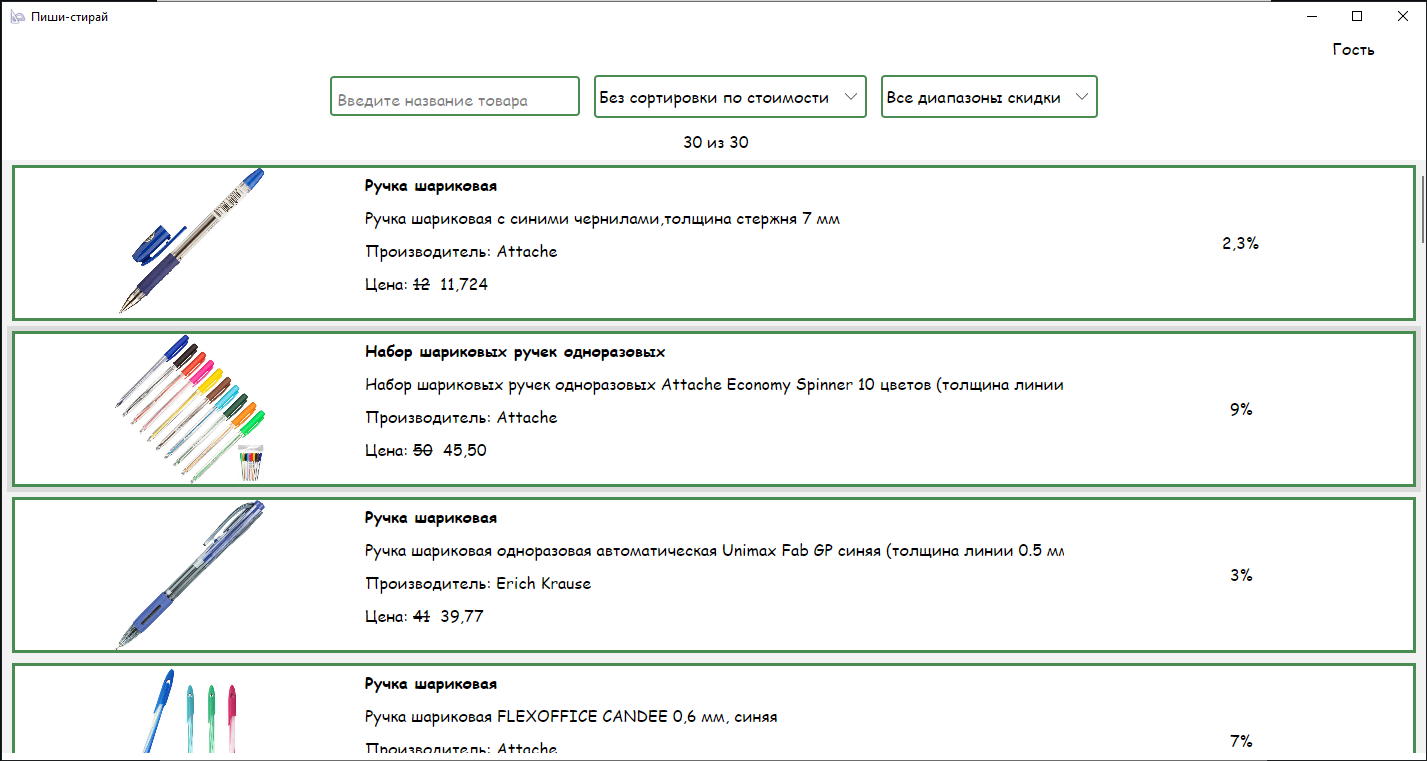


Рисунок 6. Страница неавторизированного пользователя.

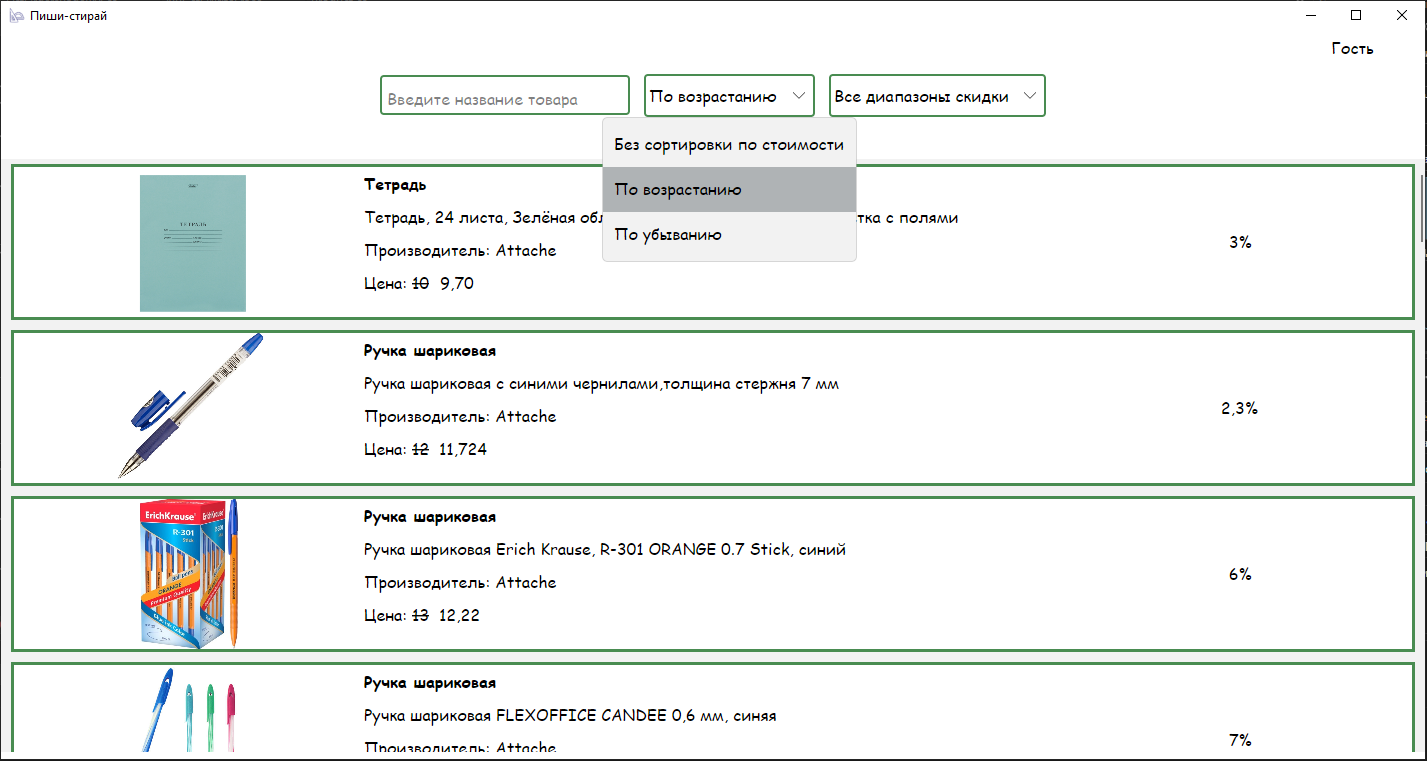


Рисунок 7. Сортировка по стоимости

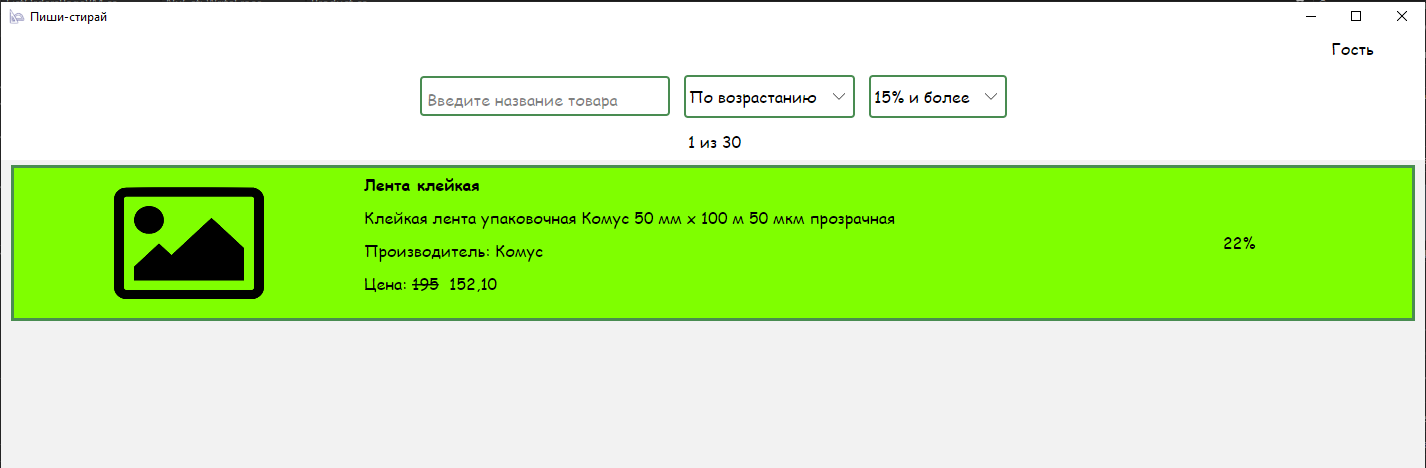


Рисунок 8. Фильтрация по скидке.

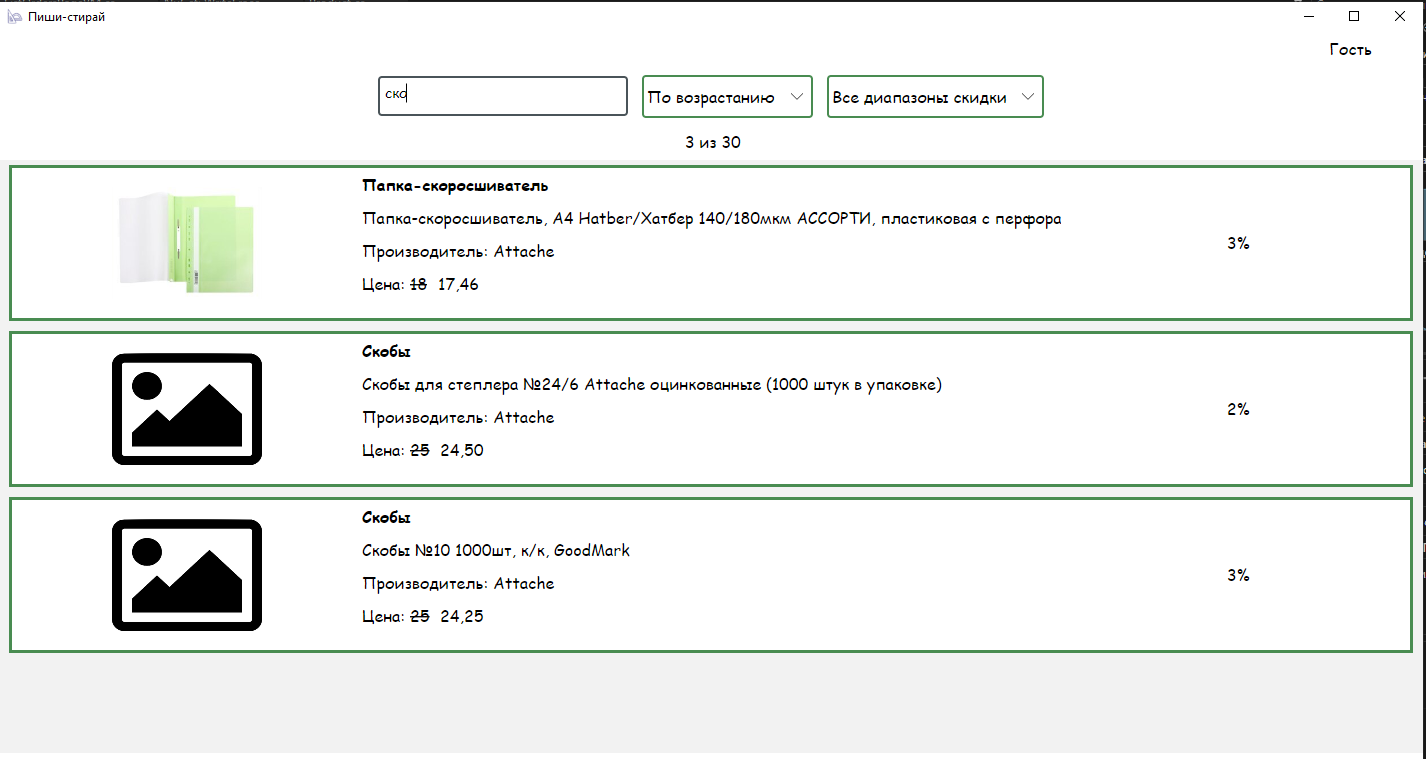


Рисунок 9. Поиск по наименованию

Также, выводится ФИО авторизированного пользователя, или, если он не авторизирован (заходит с терминала) – пишется «Гость». Для того, чтобы выйти из аккаунта необходимо нажать ПКМ на ФИО/ «Гость» и нажать на кнопку «Выйти».

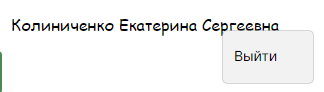


Рисунок 10. Клик ПКМ по ФИО авторизированного пользователя.

1. Формирование заказа

Реализована возможность формирования заказа как для авторизованных пользователей, так и для гостей. Пользователь может добавить товар в заказ через контекстное меню.

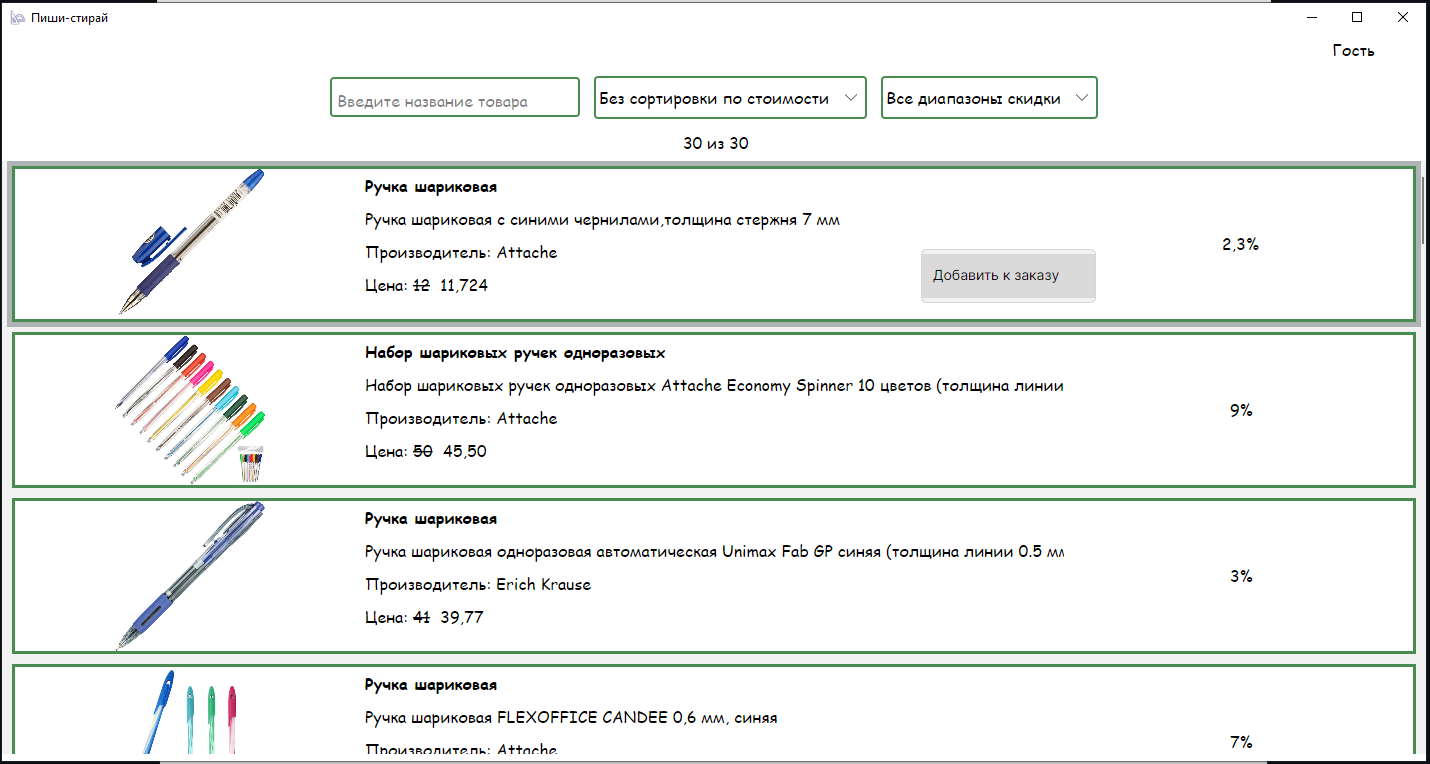


Рисунок 11. Контекстное меню

При добавлении хотя бы одного товара появляется кнопка для просмотра заказа.

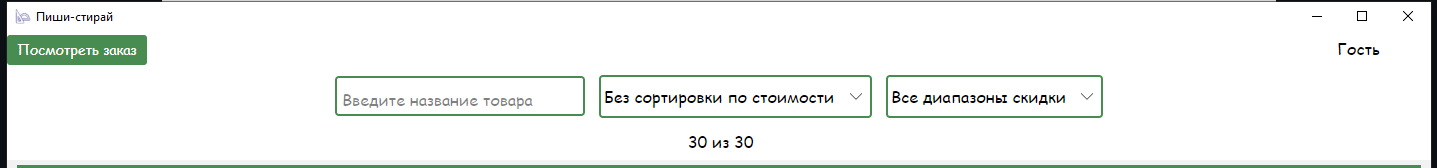


Рисунок 12. Кнопка просмотра заказа.

Заказ отображается в отдельном окне с возможностью удаления товаров. В реальном времени ведется подсчет суммы заказа и скидки. Заказу автоматически присваивается номер, и информация о заказе сохраняется в базе данных. Реализована возможность выбора пункта выдачи и формирования талона с сохранением в PDF.

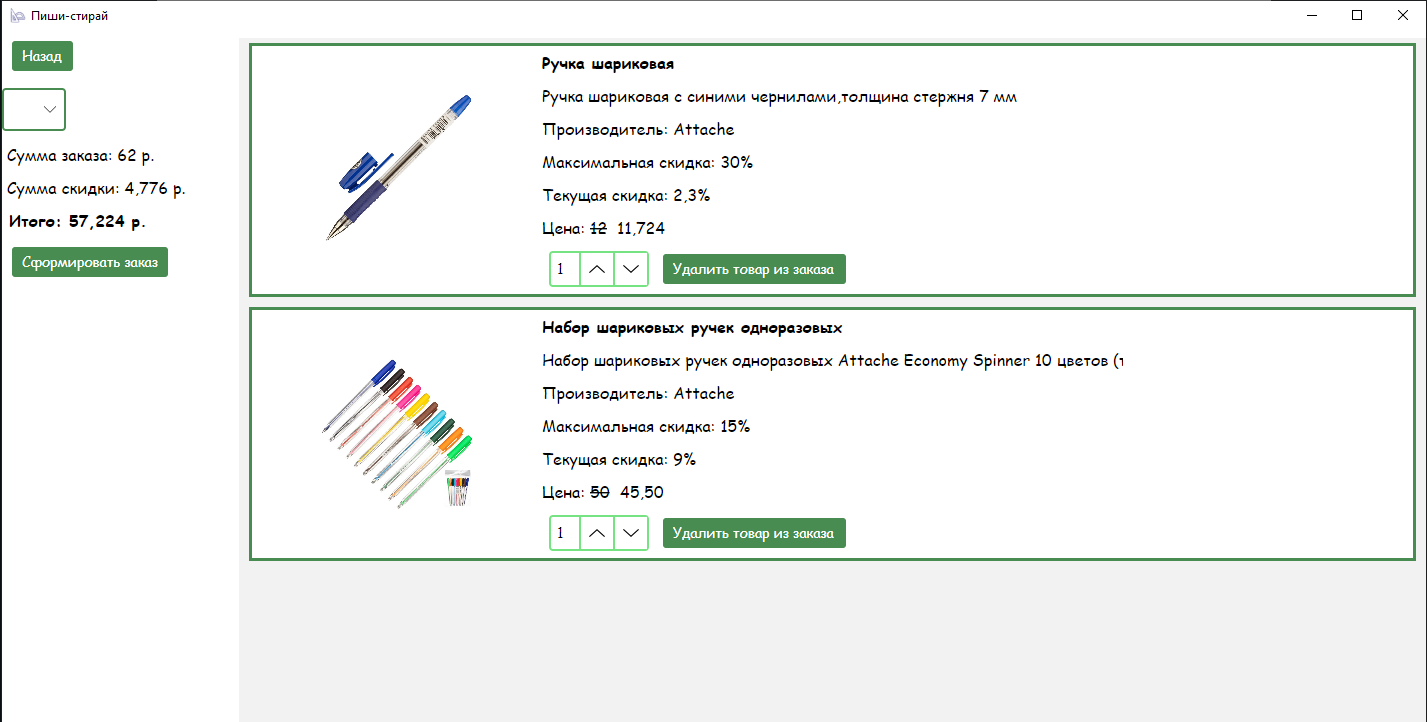


Рисунок 13. Окно формирования заказа.

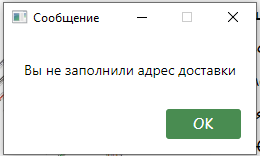


Рисунок 14. Ошибка заполнения адреса доставки.

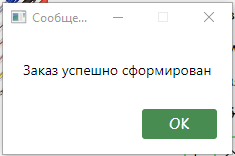


Рисунок 15. Успешность оформления заказа.

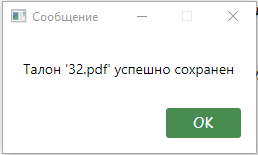


Рисунок 16. Уведомление об успешном формировании талона.

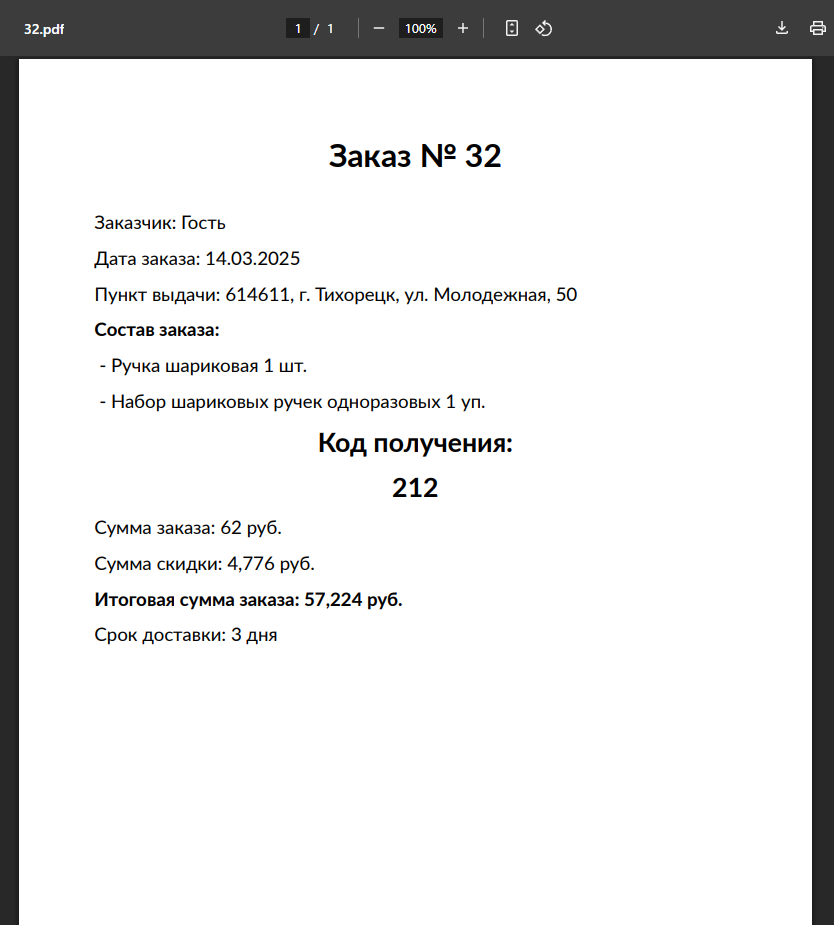


Рисунок 17. Талон заказа в формате PDF.

1. Работа с заказами

Реализован интерфейс для работы с заказами, доступный только менеджерам или администраторам. В интерфейсе отображаются все данные по заказу, включая состав, дату, сумму и скидку.

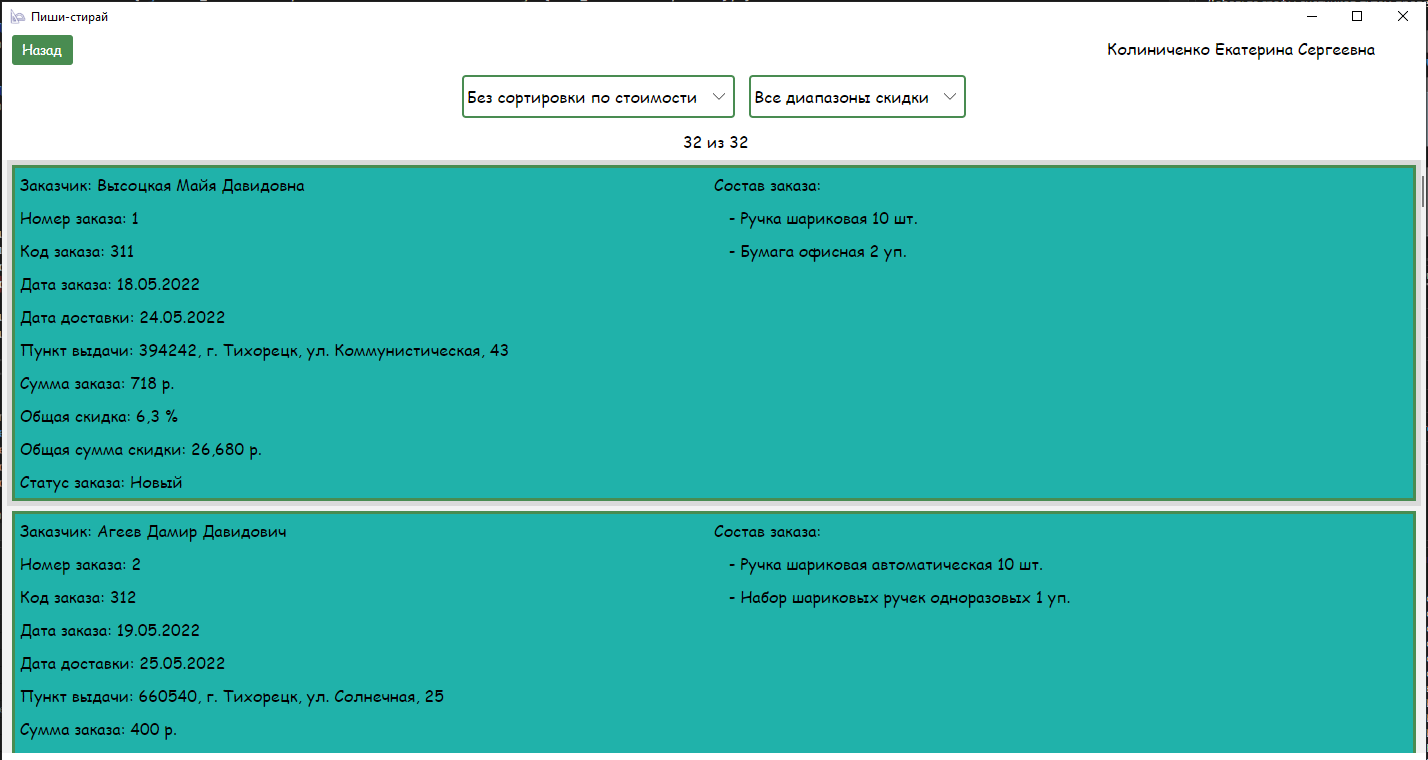


Рисунок 18. Окно работы с заказами

Реализованы функции сортировки по стоимости и фильтрации по размеру скидки. Строки с заказами выделяются цветом в зависимости от наличия товаров на складе. Все изменения применяются в реальном времени. Также, есть кнопка вернуться назад, которая возвращает пользователя на экран просмотра списка продуктов. Также, отображается количество записей и ФИО пользователя, откуда он также, может выйти.

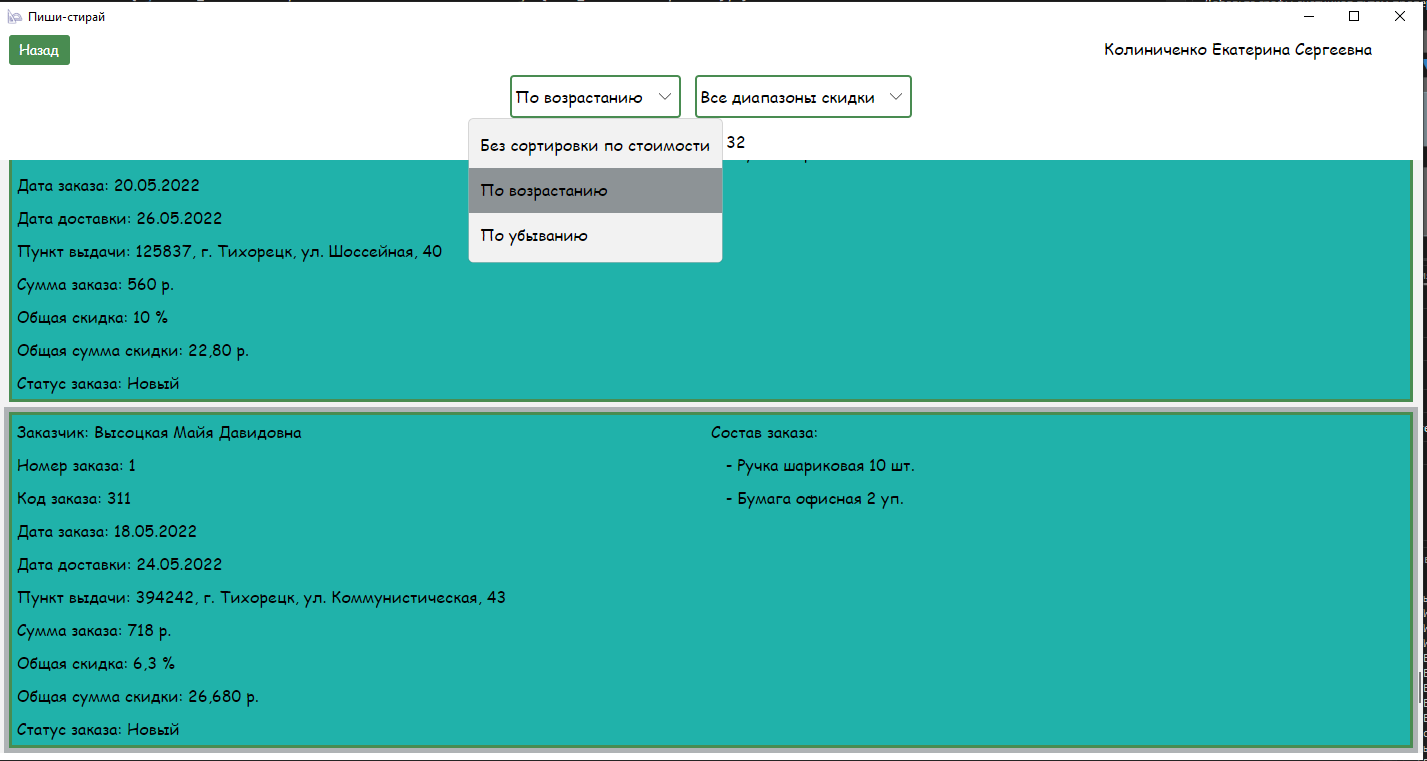


Рисунок 19. Сортировка по стоимости

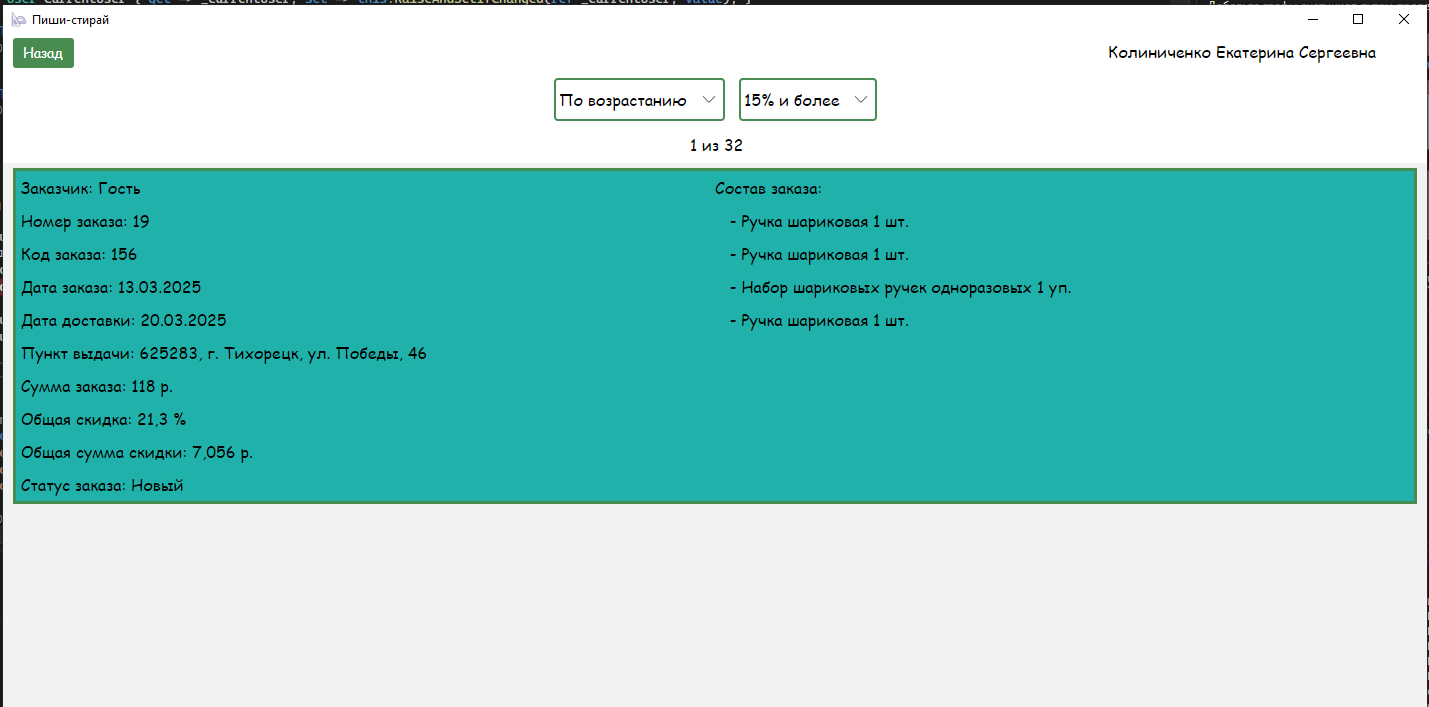


Рисунок 20. Фильтрация по размеру скидки.

## **Модуль 3. Разработка библиотеки**

В рамках проекта была разработана специализированная библиотека, реализованная в виде отдельного модуля, для определения свободных временных интервалов в графике сотрудника. Библиотека соответствует установленным стандартам именования и форматов, включая название библиотеки, класса и метода. Метод принимает на вход данные о занятых промежутках времени, границах рабочего дня и минимальном необходимом времени для работы, возвращая массив строк, представляющих доступные временные интервалы в формате HH:mm-HH:mm. Данное решение обеспечивает эффективное планирование рабочего времени с учетом занятости сотрудника и заданных ограничений. Листинг главного метода и класса предоставлен в [Приложении Б, листинг 1](#_Приложение_Б).

## **Модуль 4. Тестирование программных решений.**

Реализовано 10 unit-тестов для библиотеки на основе технологии TDD. Тестовые данные охватывают различные сценарии:

* AvailablePeriods\_NoBusyIntervals\_ReturnsFullDay() - нет занятых промежутков
* AvailablePeriods\_BusyIntervalAtStart\_ReturnsIntervalsAfterBusyTime() - занятый промежуток в начале рабочего дня
* AvailablePeriods\_BusyIntervalAtEnd\_ReturnsIntervalsBeforeBusyTime() - занятый промежуток в конце рабочего дня
* AvailablePeriods\_NotEnoughTimeBetweenIntervals\_ReturnsValidIntervals() - недостаточно времени для консультации
* AvailablePeriods\_ConsultationTime60Minutes\_ReturnsValidIntervals() - консультация длительностью в 1 час
* AvailablePeriods\_OverlappingIntervals\_ReturnsValidIntervals() - занятые промежутки перекрываются
* AvailablePeriods\_EndWorkingTimeEarlier\_ReturnsValidIntervals() - рабочий день заканчивается раньше
* AvailablePeriods\_BeginWorkingTimeLater\_ReturnsValidIntervals() - рабочий день начинается позже
* AvailablePeriods\_ConsultationTime15Minutes\_ReturnsValidIntervals() - консультация длительностью 15 минут
* AvailablePeriods\_AllIntervalsBusy\_ReturnsEmptyList() - все промежутки заняты

Подробнее ознакомиться с содержанием Unit-тестов можно в [Приложении Б, листинг №2](#_Приложение_Б).

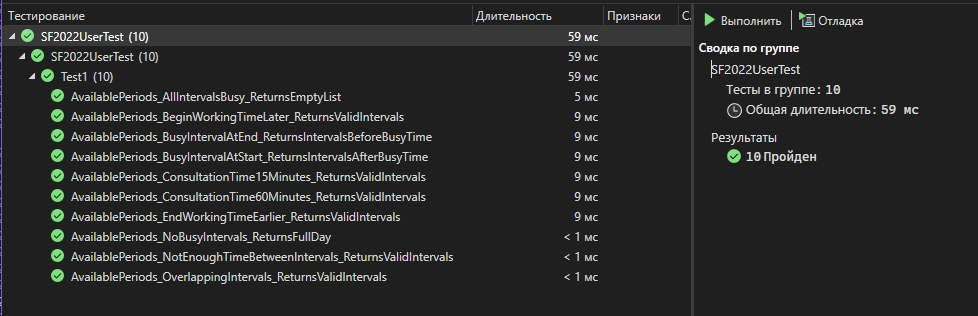


Рисунок 21. Результаты прохождения Unit-тестов.

Также описано пять сценариев тестирования добавления товара администратором:

Аннотация теста

|  |  |
| --- | --- |
| **Название проекта** | WriteErace |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Колиниченко Екатерина |
| **Дата(ы) теста** | 14. 03. 2025 |

Тестовый пример #1:

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_FUNC\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Успешное добавление товара с корректными данными |
| **Краткое изложение теста** | Проверка успешного добавления товара с корректными данными, включая выбор категории, поставщика и производителя из выпадающих списков. |
| **Этапы теста** | 1. Кликнуть на кнопку «Добавить товар» 2. Ввести корректные данные: 3. Выбрать из выпадающих списков категорию, поставщика, производителя. 4. Нажать кнопку «Добавить» |
| **Тестовые данные** | * Артикул: 12345 * Название товара: Ручка шариковая * Единица измерения: шт. * Стоимость: 50 * Максимальная скидка: 10 * Текущая скидка: 5 * Количество: 100 * Описание: Ручка шариковая синяя * Изображение: pen\_blue.jpg * Категория: Ручки * Поставщик: Комус * Производитель: Pilot |
| **Ожидаемый результат** | Товар успешно добавлен в систему. Появляется сообщение: "Товар успешно добавлен". |
| **Фактический результат** | Товар добавлен, сообщение отображается. |
| **Статус** | Зачет |
| **Предварительное условие** | 1. Приложение запущено 2. Администратор авторизован в системе. 3. В системе существуют категории, поставщики и производители. |
| **Постусловие** | Товар отображается в списке товаров. |
| **Примечания/комментарии** | - |

Тестовый пример #2:

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_FUNC\_2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Попытка добавления товара с пустым полем "Название товара" |
| **Краткое изложение теста** | Проверка обработки ошибки при попытке добавления товара с пустым полем "Название товара". |
| **Этапы теста** | 1. Кликнуть на кнопку «Добавить товар» 2. Ввести корректные данные: 3. Выбрать из выпадающих списков категорию, поставщика, производителя. 4. Нажать кнопку «Добавить» |
| **Тестовые данные** | * Артикул: 12345 * Название товара: * Единица измерения: шт. * Стоимость: 50 * Максимальная скидка: 10 * Текущая скидка: 5 * Количество: 100 * Описание: Ручка шариковая синяя * Изображение: pen\_blue.jpg * Категория: Ручки * Поставщик: Комус * Производитель: Pilot |
| **Ожидаемый результат** | Появляется сообщение об ошибке: "Поле 'Название товара' обязательно для заполнения". |
| **Фактический результат** | Сообщение об ошибке отображается. |
| **Статус** | Зачет |
| **Предварительное условие** | 1. Приложение запущено 2. Администратор авторизован в системе. 3. В системе существуют категории, поставщики и производители. |
| **Постусловие** | Товар не добавлен в систему. |
| **Примечания/комментарии** | - |

Тестовый пример #3:

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_FUNC\_3 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Попытка добавления товара с некорректной стоимостью (отрицательное значение) |
| **Краткое изложение теста** | Проверка обработки ошибки при попытке добавления товара с отрицательной стоимостью. |
| **Этапы теста** | 1. Кликнуть на кнопку «Добавить товар» 2. Ввести корректные данные: 3. Выбрать из выпадающих списков категорию, поставщика, производителя. 4. Нажать кнопку «Добавить» |
| **Тестовые данные** | * Артикул: 12345 * Название товара: Ручка шариковая * Единица измерения: шт. * Стоимость: -50 * Максимальная скидка: 10 * Текущая скидка: 5 * Количество: 100 * Описание: Ручка шариковая синяя * Изображение: pen\_blue.jpg * Категория: Ручки * Поставщик: Комус * Производитель: Pilot |
| **Ожидаемый результат** | Появляется сообщение об ошибке: "Стоимость не может быть отрицательной". |
| **Фактический результат** | Сообщение об ошибке отображается. |
| **Статус** | Зачтен |
| **Предварительное условие** | 1. Приложение запущено 2. Администратор авторизован в системе. 3. В системе существуют категории, поставщики и производители. |
| **Постусловие** | Товар не добавлен в систему. |
| **Примечания/комментарии** | - |

Тестовый пример #4:

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_FUNC\_4 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Попытка добавления товара с текущей скидкой больше максимальной |
| **Краткое изложение теста** | Проверка обработки ошибки при попытке добавления товара с текущей скидкой, превышающей максимальную. |
| **Этапы теста** | 1. Кликнуть на кнопку «Добавить товар» 2. Ввести корректные данные: 3. Выбрать из выпадающих списков категорию, поставщика, производителя. 4. Нажать кнопку «Добавить» |
| **Тестовые данные** | * Артикул: 12345 * Название товара: Ручка шариковая * Единица измерения: шт. * Стоимость: 50 * Максимальная скидка: 10 * Текущая скидка: 15 * Количество: 100 * Описание: Ручка шариковая синяя * Изображение: pen\_blue.jpg * Категория: Ручки * Поставщик: Комус * Производитель: Pilot |
| **Ожидаемый результат** | Появляется сообщение об ошибке: "Текущая скидка не может превышать максимальную". |
| **Фактический результат** | Сообщение об ошибке отображается. |
| **Статус** | Зачтен |
| **Предварительное условие** | 1. Приложение запущено 2. Администратор авторизован в системе. 3. В системе существуют категории, поставщики и производители. |
| **Постусловие** | Товар не добавлен в систему. |
| **Примечания/комментарии** | - |

Тестовый пример #5:

|  |  |
| --- | --- |
| **Test Case #** | TC\_FUNC\_5 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Попытка добавления товара с дублирующимся артикулом |
| **Краткое изложение теста** | Проверка обработки ошибки при попытке добавления товара с артикулом, который уже существует в системе. |
| **Этапы теста** | 1. Кликнуть на кнопку «Добавить товар» 2. Ввести корректные данные: 3. Выбрать из выпадающих списков категорию, поставщика, производителя. 4. Нажать кнопку «Добавить» |
| **Тестовые данные** | * Артикул: 12345 * Название товара: Ручка шариковая * Единица измерения: шт. * Стоимость: 50 * Максимальная скидка: 10 * Текущая скидка: 15 * Количество: 100 * Описание: Ручка шариковая синяя * Изображение: pen\_blue.jpg * Категория: Ручки * Поставщик: Комус * Производитель: Pilot |
| **Ожидаемый результат** | Появляется сообщение об ошибке: "Товар с таким артикулом уже существует". |
| **Фактический результат** | Сообщение об ошибке отображается. |
| **Статус** | Зачтен |
| **Предварительное условие** | 1. Приложение запущено 2. Администратор авторизован в системе. 3. В системе существуют категории, поставщики и производители. |
| **Постусловие** | Товар не добавлен в систему |
| **Примечания/комментарии** | - |

# **ЗАКЛЮЧЕНИЕ**

В ходе учебной практики были успешно освоены общие и профессиональные компетенции, включая формирование алгоритмов разработки программных модулей, их реализацию в соответствии с техническим заданием, отладку с использованием специализированных инструментов, тестирование, рефакторинг и оптимизацию кода. Для достижения поставленной цели были выполнены ключевые задачи: создание и наполнение базы данных, разработка desktop-приложения, реализация библиотеки в соответствии с техническим заданием, а также проведение отладки и тестирования готового приложения.

Итогом практики стал отчет, содержащий подробное описание выполненных заданий, а также графические материалы, такие как диаграммы, таблицы и иллюстрации, которые наглядно демонстрируют результаты работы. Практика позволила закрепить теоретические знания и приобрести практические навыки, необходимые для дальнейшей профессиональной деятельности.

# **Приложение А**

Листинг №1 - Скрипт создания БД.

|  |
| --- |
| -- DROP TABLE public.address;  -- DROP TABLE public.category;  -- DROP TABLE public.manufacturer;  -- DROP TABLE public.roles;  -- DROP TABLE public.status;  -- DROP TABLE public.supplier;  -- DROP TABLE public.products;  -- DROP TABLE public.users;  -- DROP TABLE public.orders;  -- DROP TABLE public.order\_products;  **CREATE** **TABLE** public.address (  id serial4 **NOT** **NULL**,  "name" **varchar** **NULL**,  **CONSTRAINT** address\_pk **PRIMARY** **KEY** (id)  );  **CREATE** **TABLE** public.category (  id serial4 **NOT** **NULL**,  "name" **varchar** **NULL**,  **CONSTRAINT** category\_pk **PRIMARY** **KEY** (id)  );  **CREATE** **TABLE** public.manufacturer (  id serial4 **NOT** **NULL**,  "name" **varchar** **NULL**,  **CONSTRAINT** manufacturer\_pk **PRIMARY** **KEY** (id)  );  **CREATE** **TABLE** public.roles (  id serial4 **NOT** **NULL**,  "name" **varchar** **NOT** **NULL**,  **CONSTRAINT** roles\_pk **PRIMARY** **KEY** (id)  );  **CREATE** **TABLE** public.status (  id serial4 **NOT** **NULL**,  "name" **varchar** **NULL**,  **CONSTRAINT** status\_pk **PRIMARY** **KEY** (id)  );  **CREATE** **TABLE** public.supplier (  id **int4** **DEFAULT** **nextval**('newtable\_id\_seq'::**regclass**) **NOT** **NULL**,  "name" **varchar** **NULL**,  **CONSTRAINT** newtable\_pk **PRIMARY** **KEY** (id)  );  **CREATE** **TABLE** public.products (  article **varchar** **NOT** **NULL**,  name\_products **varchar** **NULL**,  unit **varchar** **NULL**,  "cost" **int4** **NULL**,  max\_discount **int4** **NULL**,  id\_manufacturer **int4** **NOT** **NULL**,  id\_supplier **int4** **NOT** **NULL**,  id\_category **int4** **NOT** **NULL**,  current\_discount **numeric** **NULL**,  count **int4** **NULL**,  description **varchar** **NULL**,  image **varchar** **NULL**,  **CONSTRAINT** products\_pk **PRIMARY** **KEY** (article),  **CONSTRAINT** products\_category\_fk **FOREIGN** **KEY** (id\_category) **REFERENCES** public.category(id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,  **CONSTRAINT** products\_manufacturer\_fk **FOREIGN** **KEY** (id\_manufacturer) **REFERENCES** public.manufacturer(id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,  **CONSTRAINT** products\_supplier\_fk **FOREIGN** **KEY** (id\_supplier) **REFERENCES** public.supplier(id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**  );  **CREATE** **TABLE** public.users (  id serial4 **NOT** **NULL**,  surname **varchar** **NULL**,  "name" **varchar** **NULL**,  patronymic **varchar** **NULL**,  login **varchar** **NULL**,  "password" **varchar** **NULL**,  id\_role **int4** **NOT** **NULL**,  **CONSTRAINT** users\_pk **PRIMARY** **KEY** (id),  **CONSTRAINT** users\_roles\_fk **FOREIGN** **KEY** (id\_role) **REFERENCES** public.roles(id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**  );  **CREATE** **TABLE** public.orders (  id serial4 **NOT** **NULL**,  date\_order **date** **NULL**,  date\_delivery **date** **NULL**,  id\_address **int4** **NOT** **NULL**,  code **int4** **NULL**,  id\_status **int4** **NOT** **NULL**,  id\_client **int4** **NULL**,  **CONSTRAINT** orders\_pk **PRIMARY** **KEY** (id),  **CONSTRAINT** orders\_address\_fk **FOREIGN** **KEY** (id\_address) **REFERENCES** public.address(id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,  **CONSTRAINT** orders\_status\_fk **FOREIGN** **KEY** (id\_status) **REFERENCES** public.status(id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,  **CONSTRAINT** orders\_users\_fk **FOREIGN** **KEY** (id\_client) **REFERENCES** public.users(id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**  );  **CREATE** **TABLE** public.order\_products (  id serial4 **NOT** **NULL**,  id\_order **int4** **NOT** **NULL**,  article\_products **varchar** **NOT** **NULL**,  count\_products **int4** **NOT** **NULL**,  **CONSTRAINT** order\_products\_pk **PRIMARY** **KEY** (id),  **CONSTRAINT** order\_products\_orders\_fk **FOREIGN** **KEY** (id\_order) **REFERENCES** public.orders(id),  **CONSTRAINT** order\_products\_products\_fk **FOREIGN** **KEY** (article\_products) **REFERENCES** public.products(article)  ); |

Листинг №2 – Скрипт заполнения БД.

|  |
| --- |
| **INSERT** **INTO** public.address  (id, "name")  **VALUES**(1, '344288, г. Тихорецк, ул. Чехова, 1');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(2, '614164, г.Тихорецк, ул. Степная, 30');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(3, '394242, г. Тихорецк, ул. Коммунистическая, 43');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(4, '660540, г. Тихорецк, ул. Солнечная, 25');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(5, '125837, г. Тихорецк, ул. Шоссейная, 40');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(6, '125703, г. Тихорецк, ул. Партизанская, 49');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(7, '625283, г. Тихорецк, ул. Победы, 46');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(8, '614611, г. Тихорецк, ул. Молодежная, 50');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(9, '454311, г.Тихорецк, ул. Новая, 19');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(10, '660007, г.Тихорецк, ул. Октябрьская, 19');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(11, '603036, г. Тихорецк, ул. Садовая, 4');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(12, '450983, г.Тихорецк, ул. Комсомольская, 26');  **INSERT** **INTO** public.address  (id, "name")  **VALUES**(13, '394782, г. Тихорецк, ул. Чехова, 3');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(14, '603002, г. Тихорецк, ул. Дзержинского, 28');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(15, '450558, г. Тихорецк, ул. Набережная, 30');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(16, '394060, г.Тихорецк, ул. Фрунзе, 43');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(17, '410661, г. Тихорецк, ул. Школьная, 50');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(18, '625590, г. Тихорецк, ул. Коммунистическая, 20');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(19, '625683, г. Тихорецк, ул. 8 Марта');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(20, '400562, г. Тихорецк, ул. Зеленая, 32');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(21, '614510, г. Тихорецк, ул. Маяковского, 47');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(22, '410542, г. Тихорецк, ул. Светлая, 46');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(23, '620839, г. Тихорецк, ул. Цветочная, 8');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(24, '443890, г. Тихорецк, ул. Коммунистическая, 1');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(25, '603379, г. Тихорецк, ул. Спортивная, 46');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(26, '603721, г. Тихорецк, ул. Гоголя, 41');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(27, '410172, г. Тихорецк, ул. Северная, 13');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(28, '420151, г. Тихорецк, ул. Вишневая, 32');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(29, '125061, г. Тихорецк, ул. Подгорная, 8');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(30, '630370, г. Тихорецк, ул. Шоссейная, 24');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(31, '614753, г. Тихорецк, ул. Полевая, 35');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(32, '426030, г. Тихорецк, ул. Маяковского, 44');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(33, '450375, г. Тихорецк ул. Клубная, 44');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(34, '625560, г. Тихорецк, ул. Некрасова, 12');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(35, '630201, г. Тихорецк, ул. Комсомольская, 17');  **INSERT** **INTO** public.address  (id, **"name"**)  **VALUES**(36, '190949, г. Тихорецк, ул. Мичурина, 26');  **INSERT** **INTO** public.category  (id, **"name"**)  **VALUES**(1, 'Школьные пренадлежности');  **INSERT** **INTO** public.category  (id, **"name"**)  **VALUES**(2, 'Для офиса');  **INSERT** **INTO** public.category  (id, **"name"**)  **VALUES**(3, 'Бумага офисная');  **INSERT** **INTO** public.category  (id, **"name"**)  **VALUES**(4, 'Тетради школьные');  **INSERT** **INTO** public.manufacturer  (id, **"name"**)  **VALUES**(1, 'Pilot');  **INSERT** **INTO** public.manufacturer  (id, **"name"**)  **VALUES**(2, 'Attache');  **INSERT** **INTO** public.manufacturer  (id, **"name"**)  **VALUES**(3, 'Unimax');  **INSERT** **INTO** public.manufacturer  (id, **"name"**)  **VALUES**(4, 'FLEXOFFICE CANDEE');  **INSERT** **INTO** public.manufacturer  (id, **"name"**)  **VALUES**(5, 'Svetocopy');  **INSERT** **INTO** public.manufacturer  (id, **"name"**)  **VALUES**(6, 'Hatber');  **INSERT** **INTO** public.manufacturer  (id, **"name"**)  **VALUES**(7, 'Erich Krause');  **INSERT** **INTO** public.manufacturer  (id, **"name"**)  **VALUES**(8, 'Комус');  **INSERT** **INTO** public.manufacturer  (id, **"name"**)  **VALUES**(9, 'GoodMark');  **INSERT** **INTO** public.roles  (id, "name")  **VALUES**(1, 'Клиент');  **INSERT** **INTO** public.roles  (id, "name")  **VALUES**(2, 'Администратор');  **INSERT** **INTO** public.roles  (id, "name")  **VALUES**(3, 'Менеджер');  **INSERT** **INTO** public.status  (id, "name")  **VALUES**(1, 'Новый ');  **INSERT** **INTO** public.status  (id, "name")  **VALUES**(2, 'Завершен');  **INSERT** **INTO** public.supplier  (id, "name")  **VALUES**(1, 'Комус');  **INSERT** **INTO** public.supplier  (id, "name")  **VALUES**(2, 'Буквоед');  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(1, 'Никифоров', 'Всеволод', 'Иванович', 'loginDEjrm2018', 'Cpb+Im', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(2, 'Воронов', 'Донат', 'Никитевич', 'loginDEpxl2018', 'P6h4Jq', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(3, 'Игнатьева', 'Евгения', 'Валентиновна', 'loginDEwgk2018', '&mfI9l', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(4, 'Буров', 'Федот', 'Егорович', 'loginDEpou2018', 'gX3f5Z', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(5, 'Иван', 'Семёновна', '', 'loginDEjwl2018', 'D4ZYHt', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(6, 'Денисов', 'Дамир', 'Филатович', 'loginDEabf2018', '\*Tasm+', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(7, 'Ершов', 'Максим', 'Геласьевич', 'loginDEwjm2018', 'k}DJKo', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(8, 'Копылов', 'Куприян', 'Пётрович', 'loginDEjvz2018', '&|bGTy', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(9, 'Носов', 'Валерьян', 'Дмитрьевич', 'loginDEuyv2018', '8hhrZ}', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(10, 'Силин', 'Игорь', 'Авдеевич', 'loginDExdm2018', 'DH68L9', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(11, 'Дроздова', 'Александра', 'Мартыновна', 'loginDEeiv2018', 'H\*BxlS', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(12, 'Дроздов', 'Аркадий', 'Геласьевич', 'loginDEfuc2018', 'VuM+QT', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(13, 'Боброва', 'Варвара', 'Евсеевна', 'loginDEoot2018', 'usi{aT', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(14, 'Чернова', 'Агата', 'Данииловна', 'loginDElhk2018', 'Okk0jY', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(15, 'Лыткина', 'Ульяна', 'Станиславовна', 'loginDEazg2018', 's3bb|V', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(16, 'Лаврентьев', 'Леонид', 'Игнатьевич', 'loginDEaba2018', '#ИМЯ?', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(17, 'Кулаков', 'Юрий', 'Владленович', 'loginDEtco2018', 'tTKDJB', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(18, 'Соловьёв', 'Андрей', 'Александрович', 'loginDEsyq2018', '2QbpBN', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(19, 'Корнилова', 'Марфа', 'Макаровна', 'loginDEpxi2018', '+5X&hy', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(20, 'Белоусова', 'Любовь', 'Георгьевна', 'loginDEicr2018', '3+|Sn{', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(21, 'Анисимов', 'Никита', 'Гордеевич', 'loginDEcui2018', 'Zi1Tth', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(22, 'Стрелкова', 'Фаина', 'Федосеевна', 'loginDEpxc2018', 'G+nFsv', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(23, 'Осипов', 'Евгений', 'Иванович', 'loginDEqrd2018', 'sApUbt', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(24, 'Владимирова', 'Иванна', 'Павловна', 'loginDEsso2018', '#ИМЯ?', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(25, 'Кудрявцева', 'Жанна', 'Демьяновна', 'loginDErsy2018', '{Aa6nS', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(26, 'Матвиенко', 'Яков', 'Брониславович', 'loginDEvpz2018', 'mS0UxK', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(27, 'Селезнёв', 'Егор', 'Артёмович', 'loginDEfog2018', 'glICay', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(28, 'Брагин', 'Куприян', 'Митрофанович', 'loginDEpii2018', 'Ob}RZB', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(29, 'Гордеев', 'Виктор', 'Эдуардович', 'loginDEhyk2018', '\*gN}Tc', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(30, 'Мартынов', 'Онисим', 'Брониславович', 'loginDEdxi2018', 'ywLUbA', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(31, 'Никонова', 'Евгения', 'Павловна', 'loginDEzro2018', 'B24s6o', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(32, 'Полякова', 'Анна', 'Денисовна', 'loginDEuxg2018', 'K8jui7', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(33, 'Макарова', 'Пелагея', 'Антониновна', 'loginDEllw2018', 'jNtNUr', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(34, 'Андреева', 'Анна', 'Вячеславовна', 'loginDEddg2018', 'gGGhvD', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(35, 'Кудрявцева', 'Кира', 'Ефимовна', 'loginDEpdz2018', '#ИМЯ?', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(36, 'Шилова', 'Кира', 'Егоровна', 'loginDEyiw2018', 'cnj3QR', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(37, 'Ситников', 'Игорь', 'Борисович', 'loginDEqup2018', '95AU|R', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(38, 'Русаков', 'Борис', 'Христофорович', 'loginDExil2018', 'w+++Ht', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(39, 'Капустина', 'Ульяна', 'Игоревна', 'loginDEkuv2018', 'Ade++|', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(40, 'Беляков', 'Семён', 'Германнович', 'loginDEmox2018', 'Je}9e7', 3);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(41, 'Гурьев', 'Ириней', 'Игнатьевич', 'loginDEvug2018', 'lEa{Cn', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(42, 'Мишин', 'Христофор', 'Леонидович', 'loginDEzre2018', 'N\*VX+G', 2);  **INSERT** **INTO** public.users  (id, surname, "name", patronymic, login, "password", id\_role)  **VALUES**(43, 'Лазарева', 'Антонина', 'Христофоровна', 'loginDEbes2018', 'NaVtyH', 3);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(44, 'Маркова', 'Ираида', 'Сергеевна', 'loginDEkfg2018', 'r1060q', 2);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(45, 'Носкова', 'Пелагея', 'Валерьевна', 'loginDEyek2018', 'KY2BL4', 2);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(46, 'Баранов', 'Станислав', 'Дмитрьевич', 'loginDEloq2018', 'NZV5WR', 2);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(47, 'Ефремов', 'Демьян', 'Артёмович', 'loginDEjfb2018', 'TNT+}h', 3);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(48, 'Константинов', 'Всеволод', 'Мэлсович', 'loginDEueq2018', 'GqAUZ6', 3);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(49, 'Ситникова', 'Ираида', 'Андреевна', 'loginDEpqz2018', 'F0Bp7F', 3);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(50, 'Матвеев', 'Кондрат', 'Иванович', 'loginDEovk2018', 'JyJM{A', 2);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(51, 'Высоцкая', 'Майя', 'Давидовна', 'loginDExzy2018', 'R3kL!9', 1);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(52, 'Агеев', 'Дамир', 'Давидович', 'loginDEabc2018', 'T7uV@5', 1);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(53, 'Терентьев', 'Филипп', 'Богданович', 'loginDEmno2018', 'Q2wE#4', 1);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(54, 'Голубева', 'Лея', 'Петровна', 'loginDEqrs2018', 'P8yH^1', 1);  **INSERT** **INTO** public.users  (id, surname, **"name"**, patronymic, login, **"password"**, id\_role)  **VALUES**(55, 'Колиниченко', 'Екатерина', 'Сергеевна', 'admin', 'admin', 2);  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('А112Т4', 'Ручка шариковая', 'шт.', 12, 30, 2, 1, 4, 2.3, 6, 'Ручка шариковая с синими чернилами,толщина стержня 7 мм', 'А112Т4.jpg');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('T564P5', 'Набор шариковых ручек одноразовых', 'уп.', 50, 15, 2, 1, 4, 9, 5, 'Набор шариковых ручек одноразовых Attache Economy Spinner 10 цветов (толщина линии 0.5 мм)', 'T564P5.jpg');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('B730E2', 'Ручка шариковая', 'шт.', 41, 10, 7, 1, 4, 3, 45, 'Ручка шариковая одноразовая автоматическая Unimax Fab GP синяя (толщина линии 0.5 мм)', 'B730E2.jpg');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('G278R6', 'Ручка шариковая', 'шт.', 15, 30, 2, 2, 2, 7, 23, 'Ручка шариковая FLEXOFFICE CANDEE 0,6 мм, синяя', 'G278R6.png');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('R259E6', 'Бумага офисная', 'уп.', 299, 25, 5, 2, 3, 4, 32, 'Бумага офисная Svetocopy NEW A4 80г 500л', 'R259E6.jpg');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('H452A3', 'Тетрадь', 'шт.', 10, 8, 2, 2, 4, 3, 25, 'Тетрадь, 24 листа, Зелёная обложка Hatber/Хатбер, офсет, клетка с полями', 'H452A3.png');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('A543T6', 'Ручка шариковая', 'шт.', 13, 30, 2, 2, 2, 6, 12, 'Ручка шариковая Erich Krause, R-301 ORANGE 0.7 Stick, синий', 'A543T6.jpg');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('F719R5', 'Папка-скоросшиватель', 'шт.', 18, 20, 2, 2, 2, 3, 8, 'Папка-скоросшиватель, А4 Hatber/Хатбер 140/180мкм АССОРТИ, пластиковая с перфорацией прозрачный верх', 'F719R5.jpg');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('D419T7', 'Клей-карандаш', 'шт.', 61, 18, 2, 2, 2, 4, 26, 'Клей-карандаш Erich Krause 15 гр.', 'D419T7.png');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('N592T4', 'Стикеры', 'уп.', 34, 15, 2, 1, 2, 2, 17, 'Стикеры Attache 76x76 мм пастельные желтые (1 блок, 100 листов)', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('N459R6', 'Стикеры', 'уп.', 194, 25, 2, 1, 2, 3, 9, 'Стикеры Attache Selection 51х51 мм неоновые 5 цветов (1 блок, 250 листов)', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('S276E6', 'Скрепки', 'уп.', 46, 30, 8, 1, 2, 2, 14, 'Скрепки Комус металлические никелированные 33 мм (100 штук в упаковке)', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('S453G7', 'Скрепки', 'уп.', 21, 15, 2, 1, 2, 4, 20, 'Скрепки 28 мм Attache металлические (100 штук в упаковке)', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('J539R3', 'Кнопки', 'уп.', 96, 20, 8, 1, 2, 3, 24, 'Кнопки канцелярские Комус металлические цветные (50 штук в упаковке)', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('Z539E3', 'Лента клейкая', 'шт.', 16, 15, 2, 2, 2, 2, 14, 'Лента клейкая 12мм\*33м прозрачная, Hatber/Хатбер', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('A297U6', 'Ручка гелевая', 'шт.', 52, 20, 2, 2, 4, 4, 18, 'Ручка гелевая ErichKrause® G-Cube®, цвет чернил черный', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('A567R4', 'Ручка шариковая', 'шт.', 64, 30, 2, 2, 4, 2, 32, 'Шариковая ручка PILOT SuperGrip 0,7 мм синяя BPGP-10R-F-L', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('K932R4', 'Корректирующая лента', 'шт.', 70, 25, 2, 2, 2, 3, 16, 'Корректор лента 5мм\*4м, блистер, GoodMark', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('K345R5', 'Корректирующая лента', 'шт.', 87, 20, 2, 1, 2, 3, 12, 'Корректирующая лента Attache Economy 5 мм x 5 м', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('K753R3', 'Корректирующая жидкость', 'шт.', 50, 30, 2, 1, 4, 2, 5, 'Корректирующая жидкость (штрих) Attache быстросохнущая 20 мл', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, **"cost"**, max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, **count**, description, image)  **VALUES**('S563T6', 'Степлер', 'шт.', 231, 25, 2, 1, 2, 4, 17, 'Степлер Attache 8215 до 25 листов черный', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, "cost", max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, count, description, image)  **VALUES**('S425T6', 'Скобы', 'уп.', 25, 20, 2, 1, 2, 2, 16, 'Скобы для степлера №24/6 Attache оцинкованные (1000 штук в упаковке)', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, "cost", max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, count, description, image)  **VALUES**('A340R5', 'Маркер', 'шт.', 66, 15, 2, 2, 2, 4, 27, 'Маркер перманентный GoodMark 2-х сторонний.для СD/DVD черный', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, "cost", max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, count, description, image)  **VALUES**('A384T5', 'Тетрадь', 'уп.', 87, 20, 2, 2, 4, 3, 23, 'Тетрадь, 18 листов, А5 линейка Hatber/Хатбер Серия Зеленая 10шт в блистере', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, "cost", max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, count, description, image)  **VALUES**('K502T9', 'Карандаш-корректор', 'шт.', 70, 25, 2, 2, 2, 2, 7, 'Карандаш-корректор GoodMark, морозостойкий, 8мл, металлический наконечник', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, "cost", max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, count, description, image)  **VALUES**('S512T7', 'Скобы', 'уп.', 25, 15, 2, 2, 2, 3, 32, 'Скобы №10 1000шт, к/к, GoodMark', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, "cost", max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, count, description, image)  **VALUES**('D367R4', 'Клей ПВА', 'шт.', 26, 20, 2, 2, 4, 4, 16, 'Клей ПВА 85г Hatber/Хатбер', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, "cost", max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, count, description, image)  **VALUES**('M892R4', 'Ножницы', 'шт.', 209, 15, 2, 1, 4, 5, 13, 'Ножницы 195 мм Attache с пластиковыми прорезиненными анатомическими ручками бирюзового/черного цвета', '');  **INSERT** **INTO** public.products  (article, name\_products, unit, "cost", max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, count, description, image)  **VALUES**('A346R4', 'Ручка шариковая автоматическая', 'шт.', 35, 20, 2, 1, 4, 5, 23, 'Ручка шариковая автоматическая с синими чернилами, диаметр шарика 0,9 мм', 'A346R4.jpg');  **INSERT** **INTO** public.products  (article, name\_products, unit, "cost", max\_discount, id\_manufacturer, id\_supplier, id\_category, current\_discount, count, description, image)  **VALUES**('Z390R4', 'Лента клейкая', 'шт.', 195, 20, 8, 1, 2, 22, 9, 'Клейкая лента упаковочная Комус 50 мм x 100 м 50 мкм прозрачная', '');  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(3, '2022-05-20', '2022-05-26', 5, 313, 1, **NULL**);  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(4, '2022-05-21', '2022-05-27', 6, 314, 1, **NULL**);  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(5, '2022-05-22', '2022-05-28', 7, 315, 2, **NULL**);  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(6, '2022-05-23', '2022-05-29', 10, 316, 1, **NULL**);  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(7, '2022-05-24', '2022-05-30', 11, 317, 1, **NULL**);  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(10, '2022-05-27', '2022-06-02', 33, 320, 1, **NULL**);  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(1, '2022-05-18', '2022-05-24', 3, 311, 1, 51);  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(2, '2022-05-19', '2022-05-25', 4, 312, 2, 52);  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(8, '2022-05-25', '2022-05-31', 20, 318, 1, 53);  **INSERT** **INTO** public.orders  (id, date\_order, date\_delivery, id\_address, code, id\_status, id\_client)  **VALUES**(9, '2022-05-26', '2022-06-01', 30, 319, 1, 54);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(1, 1, 'А112Т4', 10);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(11, 1, 'R259E6', 2);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(2, 2, 'A346R4', 10);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(12, 2, 'T564P5', 1);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(3, 3, 'B730E2', 10);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(13, 3, 'G278R6', 10);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(4, 4, 'H452A3', 10);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(14, 4, 'A543T6', 10);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(5, 5, 'F719R5', 1);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(15, 5, 'D419T7', 1);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(6, 6, 'N459R6', 2);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(16, 6, 'S276E6', 15);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(7, 7, 'J539R3', 20);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(17, 7, 'Z539E3', 1);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(8, 8, 'A567R4', 5);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(18, 8, 'K932R4', 1);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(9, 9, 'K753R3', 1);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(19, 9, 'S563T6', 1);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(10, 10, 'S425T6', 5);  **INSERT** **INTO** public.order\_products  (id, id\_order, article\_products, count\_products)  **VALUES**(20, 10, 'A340R5', 3); |

# **Приложение Б**

Листинг №1 - Код библиотеки.

|  |
| --- |
| public class Calculations  {  public string[] AvailablePeriods(TimeSpan[] startTimes, int[] durations, TimeSpan beginWorkingTime, TimeSpan endWorkingTime, int consultationTime)  {  List<string> availablePeriods = new List<string>();  // Преобразуем занятые промежутки в список интервалов  List<(TimeSpan Start, TimeSpan End)> busyPeriods = new List<(TimeSpan, TimeSpan)>();  for (int i = 0; i < startTimes.Length; i++)  {  busyPeriods.Add((startTimes[i], startTimes[i].Add(TimeSpan.FromMinutes(durations[i]))));  }  // Сортируем занятые промежутки по времени начала  busyPeriods.Sort((a, b) => a.Start.CompareTo(b.Start));  // Начинаем с начала рабочего дня  TimeSpan currentTime = beginWorkingTime;  // Перебираем занятые промежутки  foreach (var busyPeriod in busyPeriods)  {  // Если текущее время меньше начала занятого промежутка, добавляем свободные интервалы  if (currentTime < busyPeriod.Start)  {  AddAvailablePeriods(currentTime, busyPeriod.Start, consultationTime, availablePeriods);  }  // Перемещаем текущее время на конец занятого промежутка  currentTime = busyPeriod.End;  // Если текущее время вышло за пределы рабочего дня, прерываем цикл  if (currentTime >= endWorkingTime)  break;  }  // Добавляем оставшиеся свободные интервалы до конца рабочего дня  if (currentTime < endWorkingTime)  {  AddAvailablePeriods(currentTime, endWorkingTime, consultationTime, availablePeriods);  }  return availablePeriods.ToArray();  }  private void AddAvailablePeriods(TimeSpan start, TimeSpan end, int consultationTime, List<string> availablePeriods)  {  TimeSpan current = start;  while (current.Add(TimeSpan.FromMinutes(consultationTime)) <= end)  {  availablePeriods.Add($"{current:hh\\:mm}-{current.Add(TimeSpan.FromMinutes(consultationTime)):hh\\:mm}");  current = current.Add(TimeSpan.FromMinutes(consultationTime));  }  }  } |

Таблица № 9 - Код Unit-тестов к разработанной библиотеке.

|  |
| --- |
| using SF2022UserLib;  namespace SF2022UserTest  {  [TestClass]  public sealed class Test1  {  /// <summary>  /// Нет занятых промежутков  /// </summary>  [TestMethod]  public void AvailablePeriods\_NoBusyIntervals\_ReturnsFullDay()  {  // Arrange  var startTimes = new TimeSpan[] { };  var durations = new int[] { };  var beginWorkingTime = new TimeSpan(8, 0, 0);  var endWorkingTime = new TimeSpan(18, 0, 0);  int consultationTime = 30;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  Assert.AreEqual(20, result.Length); // 10 часов \* 2 интервала в час  CollectionAssert.Contains(result, "08:00-08:30");  CollectionAssert.Contains(result, "17:30-18:00");  }  /// <summary>  /// Занятый промежуток в начале рабочего дня  /// </summary>  [TestMethod]  public void AvailablePeriods\_BusyIntervalAtStart\_ReturnsIntervalsAfterBusyTime()  {  // Arrange  var startTimes = new TimeSpan[] { new TimeSpan(8, 0, 0) };  var durations = new int[] { 60 };  var beginWorkingTime = new TimeSpan(8, 0, 0);  var endWorkingTime = new TimeSpan(18, 0, 0);  int consultationTime = 30;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  CollectionAssert.Contains(result, "09:00-09:30");  CollectionAssert.DoesNotContain(result, "08:00-08:30");  }  /// <summary>  /// Занятый промежуток в конце рабочего дня  /// </summary>  [TestMethod]  public void AvailablePeriods\_BusyIntervalAtEnd\_ReturnsIntervalsBeforeBusyTime()  {  // Arrange  var startTimes = new TimeSpan[] { new TimeSpan(17, 0, 0) };  var durations = new int[] { 60 };  var beginWorkingTime = new TimeSpan(8, 0, 0);  var endWorkingTime = new TimeSpan(18, 0, 0);  int consultationTime = 30;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  CollectionAssert.Contains(result, "16:30-17:00");  CollectionAssert.DoesNotContain(result, "17:30-18:00");  }  /// <summary>  /// Недостаточно времени для консультации  /// </summary>  [TestMethod]  public void AvailablePeriods\_NotEnoughTimeBetweenIntervals\_ReturnsValidIntervals()  {  // Arrange  var startTimes = new TimeSpan[] { new TimeSpan(10, 0, 0), new TimeSpan(10, 45, 0) };  var durations = new int[] { 30, 30 };  var beginWorkingTime = new TimeSpan(8, 0, 0);  var endWorkingTime = new TimeSpan(18, 0, 0);  int consultationTime = 30;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  CollectionAssert.DoesNotContain(result, "10:30-11:00");  }  /// <summary>  /// Консультация длительностью 1 час  /// </summary>  [TestMethod]  public void AvailablePeriods\_ConsultationTime60Minutes\_ReturnsValidIntervals()  {  // Arrange  var startTimes = new TimeSpan[] { new TimeSpan(10, 0, 0) };  var durations = new int[] { 60 };  var beginWorkingTime = new TimeSpan(8, 0, 0);  var endWorkingTime = new TimeSpan(18, 0, 0);  int consultationTime = 60;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  CollectionAssert.Contains(result, "08:00-09:00");  CollectionAssert.Contains(result, "11:00-12:00");  }  /// <summary>  /// Занятые промежутки перекрываются  /// </summary>  [TestMethod]  public void AvailablePeriods\_OverlappingIntervals\_ReturnsValidIntervals()  {  // Arrange  var startTimes = new TimeSpan[] { new TimeSpan(10, 0, 0), new TimeSpan(10, 30, 0) };  var durations = new int[] { 60, 30 };  var beginWorkingTime = new TimeSpan(8, 0, 0);  var endWorkingTime = new TimeSpan(18, 0, 0);  int consultationTime = 30;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  CollectionAssert.DoesNotContain(result, "10:30-11:00");  }  /// <summary>  /// Рабочий день заканчивается раньше  /// </summary>  [TestMethod]  public void AvailablePeriods\_EndWorkingTimeEarlier\_ReturnsValidIntervals()  {  // Arrange  var startTimes = new TimeSpan[] { new TimeSpan(16, 0, 0) };  var durations = new int[] { 60 };  var beginWorkingTime = new TimeSpan(8, 0, 0);  var endWorkingTime = new TimeSpan(17, 0, 0);  int consultationTime = 30;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  CollectionAssert.Contains(result, "15:00-15:30");  CollectionAssert.DoesNotContain(result, "16:30-17:00");  }  /// <summary>  /// Рабочий день начинается позже  /// </summary>  [TestMethod]  public void AvailablePeriods\_BeginWorkingTimeLater\_ReturnsValidIntervals()  {  // Arrange  var startTimes = new TimeSpan[] { new TimeSpan(12, 0, 0) };  var durations = new int[] { 60 };  var beginWorkingTime = new TimeSpan(10, 0, 0);  var endWorkingTime = new TimeSpan(18, 0, 0);  int consultationTime = 30;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  CollectionAssert.Contains(result, "10:00-10:30");  CollectionAssert.Contains(result, "11:30-12:00");  }  /// <summary>  /// Консультация длительностью 15 минут  /// </summary>  [TestMethod]  public void AvailablePeriods\_ConsultationTime15Minutes\_ReturnsValidIntervals()  {  // Arrange  var startTimes = new TimeSpan[] { new TimeSpan(10, 0, 0) };  var durations = new int[] { 60 };  var beginWorkingTime = new TimeSpan(8, 0, 0);  var endWorkingTime = new TimeSpan(18, 0, 0);  int consultationTime = 15;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  CollectionAssert.Contains(result, "08:00-08:15");  CollectionAssert.Contains(result, "11:00-11:15");  }  /// <summary>  /// Все промежутки заняты  /// </summary>  [TestMethod]  public void AvailablePeriods\_AllIntervalsBusy\_ReturnsEmptyList()  {  // Arrange  var startTimes = new TimeSpan[] { new TimeSpan(8, 0, 0) };  var durations = new int[] { 600 };  var beginWorkingTime = new TimeSpan(8, 0, 0);  var endWorkingTime = new TimeSpan(18, 0, 0);  int consultationTime = 30;  var calculations = new Calculations();  // Act  var result = calculations.AvailablePeriods(startTimes, durations, beginWorkingTime, endWorkingTime, consultationTime);  // Assert  Assert.AreEqual(0, result.Length);  }  }  } |