Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

ПРОГРАММНОЕ СРЕДСТВО МОНИТОРИНГА И ОТОБРАЖЕНИЯ СЕТЕВОЙ СТРУКТУРЫ ОТГРУЗКИ ТОВАРОВ СО СКЛАДОВ В МАГАЗИНЫ

БГУИР КР 6-05-0612-01 122 ПЗ

Студент: гр. 451001 Соболь Н.Г.

Руководитель:

асс. Фадеева Е.Е.

Минск 2025

       Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ПОИТ

–––––––––––– Лапицкая Н.В.

(подпись)

11.02.2025

ЗАДАНИЕ

по курсовому проектированию

Студенту  Соболю Никите Глебовичу 451001

1. Тема работы Программное средство мониторинга и отображения сетевой структуры отгрузки товаров со складов в магазины.

2. Срок сдачи студентом законченной работы 02.06.2025 г.

3. Исходные данные к работе язык программирования Delphi. Программное средство на вход получат данные о товарах, складах, магазинах, транспортных маршрутах, заказах. На выход отдаёт графическое отображение сетевой структуры отгрузок, отчеты и аналитику по отгрузке товаров, уровень запасов в магазинах и на складах.

4. Содержание расчётно-пояснительной записки (перечень вопросов, которые подлежат разработке)

Введение.

1. Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству;

2. Анализ требований к программному средству и разработка функциональных требований;

3. Проектирование программного средства;

4. Создание (конструирование) программного средства;

5. Тестирование, проверка работоспособности и анализ полученных результатов;

6. Руководство по установке и использованию;

Список используемой литературы

Заключение

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

1. "Программное средство мониторинга и отображения сетевой структуры отгрузки товаров со складов в магазины", А1, схема программы, чертеж

6. Консультант по курсовой работе

Фадеева Е.Е.

7. Дата выдачи задания 11.02.2025

8. Календарный график работы над курсовой работой на весь период проектирования (с обозначением сроков выполнения и процентом от общего объёма работы):

раздел 1 к 24.02.2025 – 15 % готовности работы;

разделы 2, 3 к 20.03.2025 – 30 % готовности работы;

разделы 4, 5 к 21.04.2025 – 60 % готовности работы;

раздел 6 к 19.05.2025 – 90 % готовности работы;

оформление пояснительной записки и графического материала к 02.06.2025 – 100 % готовности работы.

Защита курсовой работы с 02.06.2025 по 06.06.2025 г.––––––––––––––––––––

РУКОВОДИТЕЛЬ–––––– Е.Е.Фадеева

(подпись)

Задание принял к исполнению –– –\_\_\_\_––

(дата и подпись студента)

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 4](#_Toc136520264)

[ВВЕДЕНИЕ 6](#_Toc136520265)

[1. АНАЛИЗ ПРОТОТИПОВ, ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОМУ 7](#_Toc136520266)

[1.1. Запись правил синтаксиса языков программирования при помощи синтаксических диаграмм 7](#_Toc136520267)

[1.2. Примеры решения аналогичных задач, анализ достоинств и недостатков известных решений 8](#_Toc136520268)

[1.3. Требования к разрабатываемому программному средству 13](#_Toc136520269)

[2. АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ СРЕДСТВУ И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ 14](#_Toc136520270)

[2.1. Описание функциональности программного средства 14](#_Toc136520271)

[2.2. Спецификация функциональных требований 16](#_Toc136520272)

[3. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА 19](#_Toc136520273)

[3.1. Проектирование алгоритма взаимодействия пользователя с полотном 19](#_Toc136520274)

[3.2. Проектирование алгоритма рисования линий 23](#_Toc136520275)

[3.3. Проектирование алгоритма рисования фигуры 24](#_Toc136520276)

[3.4. Проектирование алгоритма отмены действия 25](#_Toc136520277)

[4. КОНСТРУИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА 26](#_Toc136520278)

[4.1. Структура программного средства 26](#_Toc136520279)

[4.2. Разработка модуля Data 26](#_Toc136520280)

[4.3. Разработка модуля Stack 27](#_Toc136520281)

[4.4. Разработка модуля Model 28](#_Toc136520282)

[4.5. Разработка модуля View 30](#_Toc136520283)

[4.6. Разработка модуля Main 32](#_Toc136520284)

[4.7. Разработка модуля frmViewGridLines 34](#_Toc136520285)

[5. ТЕСТИРОВАНИЕ, ПРОВЕРКА РАБОТОСПОСОБНОСТИ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ 35](#_Toc136520286)

[6. РУКОВОДСТВО ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ 42](#_Toc136520287)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 44](#_Toc136520288)

[ЗАКЛЮЧЕНИЕ 45](#_Toc136520289)

[ПРИЛОЖЕНИЕ А 46](#_Toc136520290)

[ПРИЛОЖЕНИЕ Б 47](#_Toc136520291)

[Содержание модуля Main 47](#_Toc136520292)

[Содержание модуля Model 53](#_Toc136520293)

[Содержание модуля View 58](#_Toc136520294)

[Содержание модуля Stack 65](#_Toc136520295)

[Содержание модуля Data 66](#_Toc136520296)

[Содержание модуля frmViewGridLines 67](#_Toc136520297)

# ВВЕДЕНИЕ

Целью данного проекта является создание программного средства мониторинга и визуализации сетевой структуры отгрузок товаров со складов в магазины, которое позволит в реальном времени отслеживать движение грузов, формировать отчёты по остаткам и анализировать эффективность выполнения логистических операций. Основные функции будущего программного обеспечения включают обработку данных о складах, магазинах, товарах и отгрузках; автоматическую валидацию вводимых данных на корректность; хранение информации в локальных структурах, обеспечивающих быстрый поиск и обновление; а также сохранение и восстановление состояния в файлах.

При проектировании использован ряд принципов, позволяющих сочетать производительность и удобство: хранение объектов (складов и магазинов) и товаров в сбалансированных структурах данных (например, в виде декартовых деревьев), что создает возможность осуществлять большинство операций над объектами за логарифмическое время. Есть применение модулей визуализации, адаптирующих карту к любому разрешению экрана и отображающих склады и магазины условными символами бордового и синего цвета соответственно, а маршруты – в виде чёрных линий.

Для разработки выбран Delphi с применением RAD Studio, что обеспечивает быстрое создание графического интерфейса, поддержку работы на Windows и возможность расширения функционала за счёт встроенных компонентных библиотек.

В результате пользователю будет доступна система, которая в одном окне объединяет карту сети, таблицы с актуальным состоянием складов и магазинов, а также списки отгрузок с возможностью фильтрации по различным критериям. Пользователь получит интуитивно понятный интерфейс, позволяющий быстро создавать, редактировать и удалять склады, магазины, товары и отгрузки, а также видеть на карте, как изменяется логистическая сеть в реальном времени.

# АНАЛИЗ ПРОТОТИПОВ, ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОМУ

* 1. **Обзор литературы**
     1. [1] algorithmica [Электронный ресурс]. – Режим доступа: <https://ru.algorithmica.org/>
     2. e-maxx [Электронный ресурс]. – Режим доступа: <http://e-maxx.ru/algo>
     3. [3] docwiki.embarcadero [Электронный ресурс]. – Режим доступа: <https://docwiki.embarcadero.com/>

Подробное описание работы с delphi и RAD Studio. Документация по процедурам и функциям в языке delphi. Подробное описание создания оконных приложений при помощи RAD Studio.

* + 1. [4] Гаджинский, А. М. Логистика. Учебник – Режим доступа: https://www.booksite.ru/fulltext/logist/text.pdf

В учебнике собрата теория по логистике, которую можно применить в программном средстве.

* 1. **Примеры решения аналогичных задач**

Рассмотрим уже существующие программные средства, решающие задачу мониторинга и отображения сетевой структуры отгрузки товаров со складов в магазины:

1. Zabbix

Достоинства:

А) Бесплатное программное обеспечение;

Б) Мощные возможности мониторинга и уведомлений;

В) Есть API;

Г) Гибкая настройка.

Недостатки:

А) Высокий порог входа;

Б) Требуется тяжёлая настройка.

2) Nagios

Достоинства:

А) Высокая гибкость и настраиваемость;

Б) Множество плагинов.

Недостатки:

А) Сложная настройка и управление.

3) SAP ERP

Достоинства:

А) Высокая функциональность;

Б) Интеграция с другими модулями ERP.

Недостатки:

А) Высокая стоимость;

Б) Сложность внедрения.

4) Oracle NetSuite

Достоинства:

А) Облачная платформа;

Б) Простота использования.

Недостатки:

А) Ограниченные возможности кастомизации.

**1.3 Требования к проектируемому программному средству**

**1.3.1** Назначение разработки. Проектируемое программное средство предназначено для мониторинга и отображения сетевой структуры отгрузки товаров со складов в магазины в реальном времени. Оно должно обеспечивать отслеживание отгрузок, а также генерировать таблицы для анализа логистической цепочки. Система должна обеспечивать высокую степень автоматизации, снижать вероятность ошибок и ускорять принятие управленческих решений.

**1.3.2** Состав выполняемых функций:

1.3.2.1 Управление складами и магазинами:

1) Создание нового склада или магазина с вводом: название, улица, дом, корпус (опционально), общая вместимость. При сохранении проводится проверка ввода, создаётся узел в treap-структуре, вычисляются координаты X, Y (по данным пользователя) и добавляется графическое отображения объекта на карту;

2) Редактирование и удаление существующих объектов изменение адресных и ёмкостных параметров, проверка корректности данных и обновление treap-структур;

3) Отображение детальной информации об объекте: статистика по уровню занятости, списку товаров, возможностям отгрузок.

1.3.2.2 Управление товарами в рамках каждого узла. Добавление товара: ввод наименования, категории, объёма единицы (количество условных единиц) и текущего количества. Реализована авторизация по существующим категориям, проверка уникальности ключа товара в treap-структуре.

1.3.2.3 Управление отгрузками:

1) Создание новой отгрузки с выбором отправителя (склада или магазина), получателя (склад или магазин), наименования товара и количества. Система проверяет наличие достаточного запаса и обновляет поле, отвечающее за количество товара, которое необходимо отправить;

2) При создании отгрузки создаётся структура с полями: имя отгрузки, уникальный ID, указатели на отправителя и получателя, наименование товара и количество. Данные об отгрузке добавляются в связный список отгрузок и сразу же создаётся объект для визуализации отгрузки на карте.

1.3.2.4 Фильтрация объектов. Пользователь может задавать параметры фильтрации: тип объекта (склад/магазин), улица, дом, корпус, диапазон общей вместимости, диапазон текущей занятости. При применении фильтра вызывается функция, скрывающая все графические элементы объектов, не подходящих под критерии. Карта перерисовывается после каждой фильтрации.

1.3.2.5 Визуализация сетевой структуры отгрузок. Компонент TPaintBox загружает фон и рисует стрелки между координатами узлов. Если у отправителя или получателя свойство видимости отключено, соответствующая стрелка не рисуется.

1.3.2.6 Формирование таблиц:

1) Таблица остатков товаров во всех магазинах и на складах. При открытии формы данные извлекаются из treap-структур и отображаются в TStringGrid;

2) Таблица существующих отгрузок. Информация о каждой отгрузке читается из связного списка отгрузок и отображается с указанием: имя отгрузки, ID, отправитель, адрес отправителя, получатель, адрес получателя, товар, артикул, количество;

3) Таблицы выборочно по остаткам товаров в конкретном магазине или складе. Пользователь выбирает узел, и вызывается соответствующая процедура, проходящая по списку товаров данного узла и формирующая таблицу.

1.3.2.7 Сохранение данных:

1) Все данные хранятся в трёх текстовых файлах: shops.txt, warehouses.txt, shipments.txt;

2) При сохранении: для каждого склада/магазина записываются строки с полями: количество объектов, описание каждого объекта. Поля описания каждого объекта: название, улица, дом, корпус, вместимость, занятая вместимость, зарезервированное место, идентификатор, координаты, количество различных позиций, описание позиций. Каждая позиция содержит поля: название, категория, объём единицы товара, количество, идентификатор, количество, которое необходимо отправить.

3) При сохранении отгрузок формируется файл с полями: количество отгрузок, описание каждой отгрузки. Описание каждой отгрузки содержит: название, идентификатор, идентификаторы отправителя и получателя, название товара, количество.

**1.3.3** Входные данные:

1) Информация о складах: название склада, адрес (улица, дом), вместимость.

2) Информация о магазинах: название склада, адрес (улица, дом), вместимость.

3) Данные о товарах: название, категория, объём единицы товара (количество условных единиц)

4) Данные об отгрузках: название, тип отправителя, название отправителя, идентификатор отправителя, тип получателя, название получателя, идентификатор получателя, название товара, артикул товара, количество товара для отгрузки;

**1.3.4** Выходные данные:

1) Графическое отображение сетевой структуры отгрузок в виде карты;

2) Формирование таблиц по остаткам товаров во всех магазинах и на складах;

3) Формирование таблиц с существующими отгрузками;

4) Формирование таблиц по остаткам товаров в конкретном магазине или в конкретном складе.

**1.3.5** Требования к временным характеристикам:

1) Обновление данных в режиме реального времени;

2) Задержка между обновлением данных и их отображением на экране не должна превышать 1 секунды.

**1.3.6** Требования к надёжности. Программное средство должно быть устойчиво к программным ошибкам и сбоям, в том числе проверять все входные данные на корректность.

**1.3.7**  Условия эксплуатации. Программное средство должно поддерживать работу на ОС Windows.

**1.3.8** Язык и среда разработки. Язык разработки: Delphi, благодаря его возможности создания высокопроизводительных и надежных приложений с графическим интерфейсом. Среда разработки: Embarcadero RAD Studio, так как она предлагает мощные инструменты для визуального проектирования и отладки приложений на Delphi.

**1.3.9**  Дополнительные требования. Интерфейс должен быть простым и понятным для пользователей с разным уровнем технической подготовки. Это включает в себя использование понятных терминов, логичную структуру меню и подсказки для сложных операций. Для отображения сетевой структуры отгрузок необходимо использовать графические элементы (карта, линии на карте, множество панелей), которые позволяют быстро анализировать информацию.

# АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ СРЕДСТВУ И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

**2.1 Теоретический анализ, математическое обоснование и доказательства, модели технических объектов и результаты моделирования**

**2.1.1** Декартово дерево (treap, дерамида). Декартово дерево – структура данных, которое объединило в себе бинарное дерево поиска (tree) и бинарную кучу (heap). Эта структура данных хранит в себе пары (X, Y), так, что дерамида является бинарным деревом поиска по X и бинарной пирамидой по Y. Таким образом, если в текущей вершине лежит значение (X0, Y0), то в левом поддереве будут лежать элементы со значением X < X0, а в правом со значением X > X0. А также и в левом, и в правом поддереве будут располагаться Y < Y0. В дальнейшей реализации X – ключ, Y – приоритет. Приоритет будет генерироваться случайно, обеспечивая сложность работы O(logN) в среднем. В итоге декартово дерево позволяет выполнять следующие операции:

1) Insert(X, Y) – Вставка элемента в дерево. O(logN) в среднем;

2) Find(X) – Поиск элемента в дереве. O(logN) в среднем;

3) Erase(X) – Удаление элемента из дерева. O(logN) в среднем;

4) Merge(Root1, Root2) – Объединение деревьев. O(Mlog(N/M)) в среднем;

5) Split(Root, X) – Разделение одного дерева на 2 по ключу X. O(logN) в среднем.

**2.2 Описание функциональности программного средства**

**2.2.1** В системе добавление любого нового логистического объекта начинается с того, что пользователь открывает соответствующую форму и вводит его наименование, адрес (улица, дом, корпус), общую вместимость и координаты на карте. Система проверяет, что название не пустое, не нарушает правил длины и символов, и что оно ещё не встречается среди существующих записей; после этого создаётся запись в общем treap-дереве «объекты», где ей присваивается уникальный ключ и случайный приоритет. Одновременно на карте появляется круг определённого цвета (бордовый для склада, синий для магазина), размещённый в указанных координатах. При редактировании объекта пользователь выбирает его на карте, изменяет нужные поля (название, адрес, вместимость, координаты), и система снова проверяет корректность ввода: если изменилось имя, убеждается в его уникальности; если изменена вместимость, убеждается, что текущее использование не превосходит новое значение. После успешной обработки изменённая запись остаётся в том же месте treap-дерева, а круг на карте перемещается или меняет подпись. Удаление происходит лишь в том случае, если для объекта нет активных отгрузок: система просматривает список маршрутов и, если не находит ни одной связи, удаляет запись из treap, освобождает вложенные товары, и круг исчезает с карты.

Фильтрация работает так: пользователь задаёт условия по типу «склад/магазин», строковому адресу и числовым диапазонам вместимости и занятости, после чего программа обходит все узлы treap-дерева и скрывает или показывает соответствующие круги; вместе с этим анализируются маршруты, и, если объект скрыт, все линии, связанные с ним, тоже временно исчезают. Таким образом, поиск и управление объектами реализованы через единое treap-дерево, а визуализация и фильтрация происходят «на лету» без перезагрузки.

**2.2.2** Управление ассортиментом товаров. В рамках управления ассортиментом у каждого склада или магазина имеется собственный treap-список товаров. Чтобы добавить новый товар, пользователь выбирает нужный логистический объект, затем вводит название, категорию, объём одной единицы и количество. Система проверяет, что название корректно по формату и не повторяется среди товаров этого же объекта, а объём и количество являются целыми числами. После этого создаётся новая запись в вложенном treap «товаров» конкретного узла, и одновременно пересчитывается занятость пространства этого склада или магазина (к «текущей занятости» прибавляется произведение объёма на количество), а «доступная вместимость» уменьшается соответственно.

Для просмотра всех товаров в системе предусмотрен табличный вывод, в котором отображаются: объект, в котором лежит товар, тип объекта, вместимость этого объекта, количество занятого и зарезервированного места, название товара, категория товара, артикул, количество, место, занятое товаром.

**2.2.3** Планирование и исполнение отгрузок. Пользователь может создать новую отгрузку, выбрав отправителя (склад или магазин), получателя (склад или магазин), товар и количество условных единиц. Перед созданием, система проверяет, что у отправителя есть достаточное количество желаемого товара на балансе. После подтверждения отгрузка записывается в список отгрузок.

Также для каждой действующей отгрузки система визуально показывает маршрут на карте, связывая отправную и конечную точки.

**2.2.4** Пользователь может ограничить отображение объектов на карте по нескольким критериям: вид «склад» или «магазин», адресные параметры (улица, дом, корпус), а также диапазоны общей вместимости и текущей занятости. После ввода необходимых условий и нажатия «Применить» система последовательно обходит все записи treap-дерева логистических объектов. Каждый объект сравнивается с условиями фильтра: если хотя бы одно ограничение не выполняется, его круг на карте скрывается. При этом одновременно скрываются все маршруты, связанные с этим объектом, чтобы не оставалось «висячих» линий. При сбросе фильтра все круги и линии снова становятся видимыми, а карта обновляется. Такой подход обеспечивает быстрое и прозрачное управление тем, какие склады и магазины (и их маршруты) в данный момент отображаются пользователю.

**2.2.5** Визуализация логистической сети. На основной карте отображаются все склады и магазины условными символами (разный цвет для складов и магазинов). При наведении мыши на объект на карте можно увидеть его краткое описание.

Для каждой активной отгрузки на карте рисуется линия с направлением от точки отправления до точки прибытия. При наведении курсора мыши на отгрузку можно увидеть её подробные сведения: кто отправитель, кто получатель, какой товар, сколько штук было отправлено.

Вся визуализация автоматически обновляется при создании, обработке или удалении отгрузок, а также при добавлении или удалении объектов, чтобы карта всегда отражала актуальное состояние сети.

**2.2.6** Сохранение и восстановление состояния. Система поддерживает сохранение полного состояния базы данных (списки объектов, товары, отгрузки, балансы) в файлы. При следующем запуске приложения можно загрузить ранее сохранённые файлы, и система восстановит все объекты, товары, отгрузки и отобразит их на карте.

При загрузке данных прежние списки очищаются, затем последовательно восстанавливаются объекты с их товарами и балансовыми остатками, а затем восстанавливается список отгрузок с соответствующими маршрутами. После завершения загрузки карта автоматически перерисовывается, чтобы отобразить все объекты и текущие маршруты.

**2.3 Спецификация функциональных требований**

**2.3.1** Визуализация сетевой структуры отгрузок. Сетевая структура отгрузок (склады, магазины, маршруты) отображается в виде карты. Карта автоматически подстраивается под разрешение экрана пользователя.

Элементы карты:

-склады, которые обозначаются бордовыми кругами;

-магазины, которые обозначаются синими кругами;

-маршруты, которые отображаются черными линиями.

**2.3.2** Построение маршрута начинается в момент, когда система получает новую запись об отгрузке. В этот момент создаётся структура, в которой сохраняются ссылки на объекты-отправитель и -получатель, наименование товара, количество и уникальный идентификатор. После этого в список активных отгрузок добавляется только что созданная. А на карту добавляется прямая линия, соединяющая отправителя и получателя. В процессе работы, когда пользователь перемещает курсор по области карты, система для каждой линии вычисляет минимальное расстояние от указателя до отрезка по стандартной формуле «расстояние от точки до прямой». Если это расстояние оказывается меньше заранее заданного порога, рядом с курсором показывается всплывающая подсказка с подробными данными о соответствующей отгрузке. Такое поведение обеспечивает удобный интерактивный доступ к информации о любой отгрузке прямо на карте.

**2.3.3**  Добавление / редактирование / удаление объектов. Должно быть реализовано несколько функций, каждая из которых будет отвечать за добавление, редактирование или удаление определенного объекта (магазина / склада). На ввод подаётся информация о новом или уже существующем объекте, в зависимости от того, какая операция будет выполняться. Все данные проходят валидацию. Функция ничего не возвращает, а только добавляет объект в список, редактирует уже существующий объект или удаляет объект из списка.

# ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

# КОНСТРУИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

# ТЕСТИРОВАНИЕ, ПРОВЕРКА РАБОТОСПОСОБНОСТИ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

**5.1 Описание тестов, результаты тестирования**

Таблица 5.1 – Тестирование функционала программного средства

| **Специфика тестирования** | **Номер теста** | **Вводимые данные** | **Ожидаемый результат** | **Полученный результат** |
| --- | --- | --- | --- | --- |
| Запуск программы | 1 | Двойной щелчок левой кнопкой мыши по программе | Появление окна программы с картой | Тест пройден |
| Фильтрация | 2 | Нажатие на кнопку “фильтр” на главном экране | Появление окна со значениями для фильтра по центру экрана | Тест пройден |
| Создание магазина | 3 | Нажатие на кнопку "Создать магазин" и ввод имени, улицы, дома, емкости | Появление магазина на карте с соответствующими данными | Тест пройден |
| Создание склада | 4 | Нажатие на кнопку "Создать склад" и ввод имени, улицы, дома, емкости | Появление склада на карте с соответствующими данными | Тест пройден |
| Редактирование магазина | 5 | |  | | --- | |  |   Выбор магазина для редактирования, изменение данных (имя, улица, дом) | Обновление данных магазина на карте | Тест пройден |
| Редактирование склада | 6 | Выбор склада для редактирования, изменение данных (имя, улица, дом) | Обновление данных склада на карте | Тест пройден |

Продолжение таблицы 5.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Валидация данных при создании объекта | 7 | Ввод неправильных данных (недопустимые символы в имени или улице) | Появление ошибки и запрет на создание объекта | Тест пройден |
| Валидация данных при фильтрации | 8 | Ввод некорректных значений в фильтре (нечисловые значения в поле «Вместимость») | Появление сообщения об ошибке или запрет на фильтрацию с ошибочными данными | Тест пройден |
| Выделение объекта на карте | 9 | Наведение курсором по объекту (магазин/склад) на карте | Появление панели с подробной информацией о выбранном объекте | Тест пройден |
| Фильтрация объектов по типу | 10 | Установка фильтра «Только магазины» и нажатие кнопки «Применить» | Отображение только объектов типа «Магазин» на карте | Тест пройден |
| Фильтрация объектов по адресу | 11 | Ввод улицы и дома в фильтре, нажатие кнопки «Применить» | Отображение объектов, которые соответствуют фильтру по адресу | Тест пройден |
| Сброс фильтра | 12 | Нажатие на кнопку «Сброс» | Очистка всех фильтров, возврат к состоянию с отображением всех объектов | Тест пройден |

Продолжение таблицы 5.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Работа с полями фильтрации (интервал емкости) | 13 | Ввод минимального и максимального значения емкости в фильтре и нажатие кнопки «Применить» | Отображение объектов, которые соответствуют диапазону емкости | Тест пройден |
| Успешное создание объекта с уникальным именем | 14 | Ввод уникального имени для магазина/склада и нажатие кнопки «Подтвердить» | Создание нового объекта на карте с уникальным именем и соответствующими данными | Тест пройден |
| Проверка существующего имени объекта | 15 | Ввод существующего имени магазина/склада и попытка создать новый объект | Появление сообщения об ошибке и отказ в создании нового объекта | Тест пройден |
| Проверка обновления объектов после редактирования | 16 | Редактирование данных объекта (например, изменения вместимости склада) | Обновление отображаемых данных на карте и на панели информации | Тест пройден |
| Тестирование кнопки «Отмена» в процессе создания объекта | 17 | Нажатие кнопки «Отмена» в процессе создания объекта | Очистка введенных данных и возврат в начальное состояние формы | Тест пройден |
| Проверка работы кнопки “Удалить” | 18 | Нажатие на кнопку «Удалить» на объекте | Удаление объекта с карты и его исчезновение из базы данных | Тест пройден |

Продолжение таблицы 5.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Попытка создать объект с именем длиной ровно 70 символов (максимально допустимая длина) | 19 | При создании магазина ввести имя, состоящее из 70 букв «A» подряд | Магазин успешно создаётся (длина имени на границе допустимого), отображается на карте | Тест пройден |
| Сохранение и загрузка при отсутствии объектов и отгрузок | 20 | Запустить приложение сразу после установки, без создания магазинов, складов или отгрузок; нажать «Сохранить»; затем «Загрузить» | В файлах сохранения ничего не будет записано. После загрузки структура остаётся пустой: на карте нет ни одного круга, список отгрузок пуст. | Тест пройден |
| Сохранение после редактирования объектов и перезапуск с загрузкой | 21 | Создать магазин, затем отредактировать его, нажать «Сохранить», закрыть приложение, снова «Загрузить» | После первой сохранённой сессии файл объектов должен содержать магазин с измененными параметрами | Тест пройден |
| При фильтрации оставить пустыми все поля (никаких условий) и нажать «Применить» | 22 | В фильтре не вводить ничего, оба чекбокса «склад» и «магазин» отмечены, диапазоны пусты; нажать «Применить» | Отображаются все склады и магазины без изменений | Тест пройден |
| При фильтрации ввести в поле «вместимость от» больше, чем в поле «вместимость до» | 23 | В фильтре ввести «вместимость от» = 200, «вместимость до» = 100 | Применение фильтра становится невозможным, поле «вместимость до» меняет цвет на красный | Тест пройден |
| Попытка создать объект с именем длиной 71 символ | 24 | При создании магазина ввести имя из 71 символа «A» подряд | Появление сообщения об ошибке («Имя должно быть не более 70 символов») | Тест пройден |
| Ввод недопустимых символов в поле «улица» | 25 | При создании магазина ввести улицу «St@r!» | Становится невозможным нажатие кнопки «Создать», поле улицы меняет цвет на красный | Тест пройден |
| Отмена создания нового объекта | 26 | Нажать «Создать склад», ввести его данные, нажать «Отмена» | Все введённые данные сброшены, панель закрывается, склад не появляется на карте | Тест пройден |
|  | 27 |  |  | Тест пройден |
|  | 28 |  |  | Тест пройден |
|  | 29 |  |  | Тест пройден |
|  | 30 |  |  | Тест пройден |
|  | 31 |  |  | Тест пройден |
|  | 32 |  |  | Тест пройден |
|  | 33 |  |  | Тест пройден |
|  | 34 |  |  | Тест пройден |
|  | 35 |  |  | Тест пройден |
|  | 36 |  |  | Тест пройден |
|  | 37 |  |  | Тест пройден |
|  | 38 |  |  | Тест пройден |
|  | 39 |  |  | Тест пройден |
|  | 40 |  |  | Тест пройден |
|  | 41 |  |  | Тест пройден |
|  | 42 |  |  | Тест пройден |
|  | 43 |  |  | Тест пройден |
|  | 44 |  |  | Тест пройден |
| Завершение программы | 45 | Нажатие на «Х» в правом верхнем углу окна | Появление предупреждающего окна, завершение программы | Тест пройден |

Все тесты прошли успешно. Сбои в работе программы при прохождении тестов не были обнаружены. Программа полностью исправна и готова к использованию.

# РУКОВОДСТВО ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ

Чтобы начать использование программы, нужно использовать установщик. Для этого достаточно запустить файл setup.exe, который представлен на рисунке 6.1.

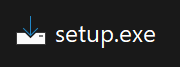


Рисунок 6.1 – Установщик программы

После запуска установщика пользователь должен выбрать режим установки в окне, представленном на рисунке 6.2.

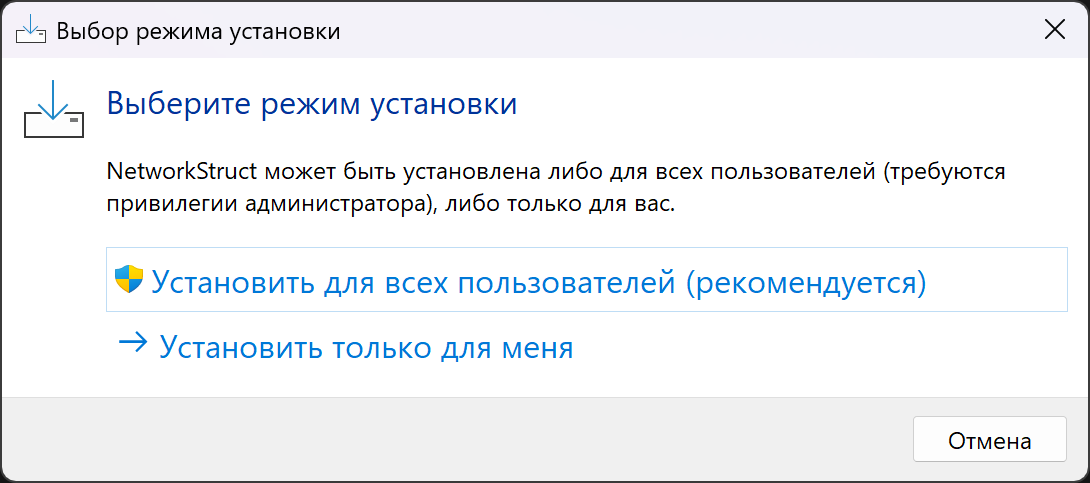


Рисунок 6.2 – Выбор режима установки

В следующем окне пользователь должен выбрать язык установщика. Выбрать можно либо русский, либо английский. Окно выбора языка установщика представлено на рисунке 6.3.

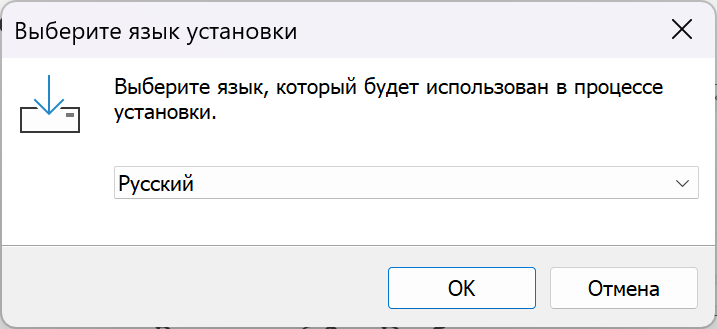


Рисунок 6.3 – Выбор языка установщика

После выбора языка пользователю предлагается выбрать место для установки приложения. Путь по умолчанию - C:\Program Files (x86)\NetworkStruct. Окно выбора пути представлено на рисунке 6.4.

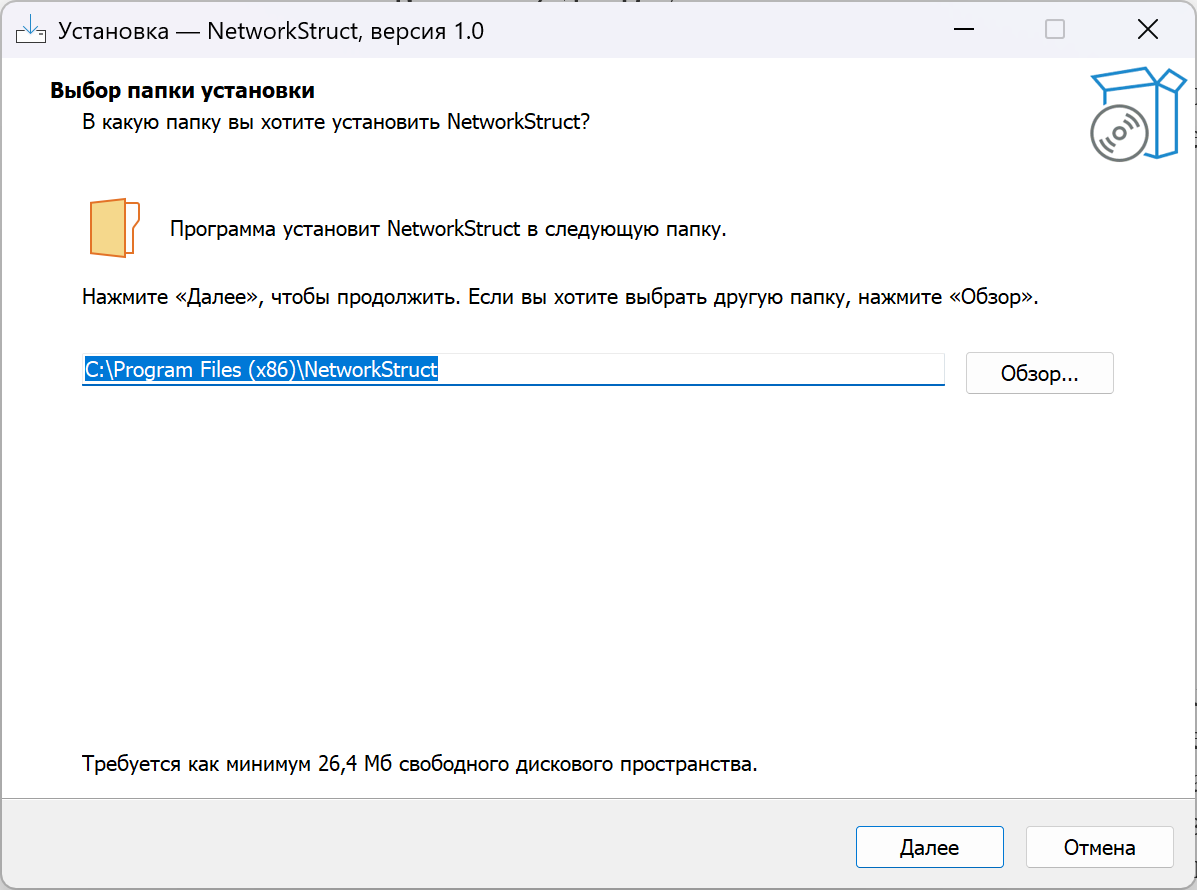


Рисунок 6.4 – Выбор пути для установки

После выбора пути открывается окно, в котором предлагается создать значок на Рабочем столе. Данное окно представлено на рисунке 6.5.

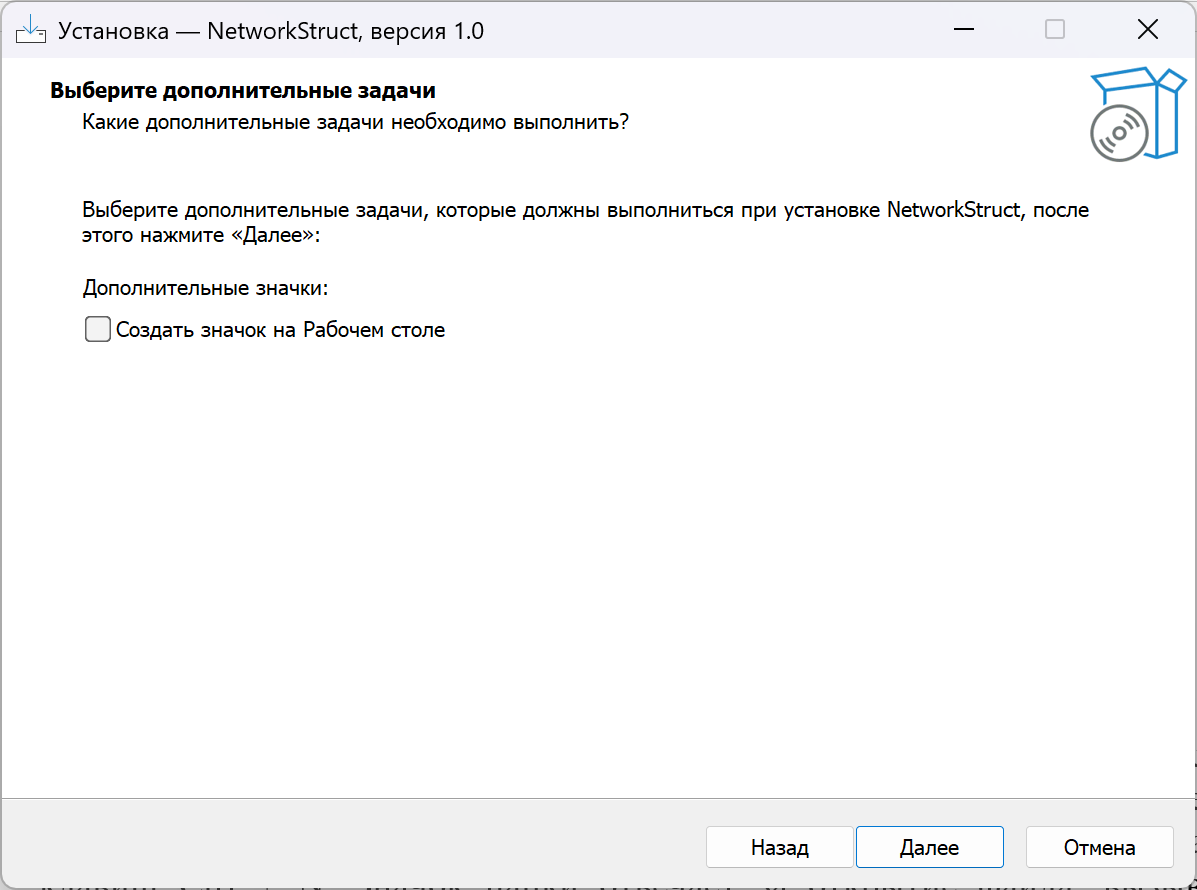


Рисунок 6.5 – Выбор дополнительных задач

Далее пользователь видит окно подтверждения установки, которое представлено на рисунке 6.6.

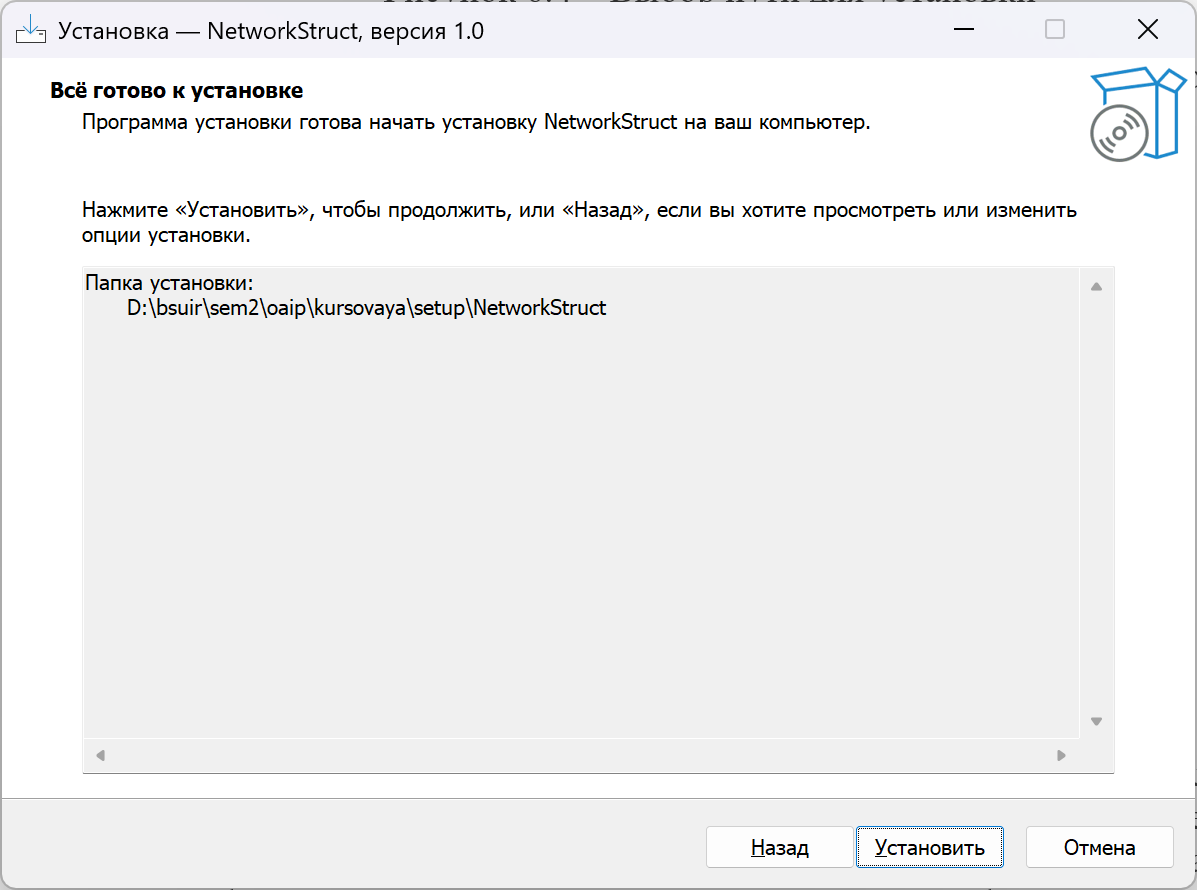


Рисунок 6.6 – Окно подтверждения установки

После установки пользователь увидит окно, представленное на рисунке 6.7, которое предложит сразу запустить программу, либо завершить установку без запуска.

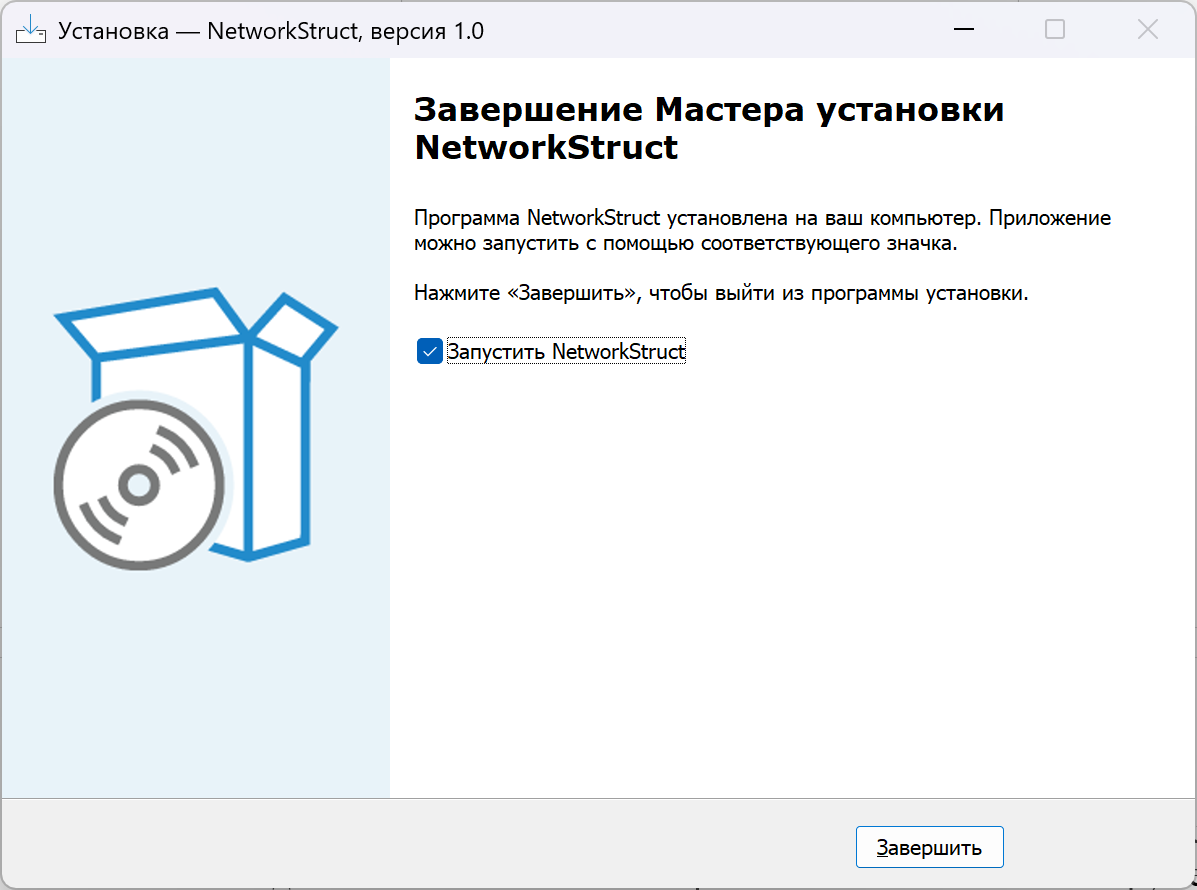


Рисунок 6.7 – Окно завершения установки

Папка, в которую была произведена установка должна содержать 7 элементов: map.bmp, Project1.exe, shipments.txt, shops.txt, unins000.dat, unins000.exe, warehouses.txt. Данная структура представлена на рисунке 6.8.

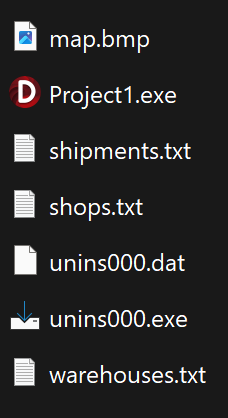


Рисунок 6.8 – Содержимое папки с программой

Map.bmp – фон (схематичное изображение карты), который отображается на главной странице программы. Project1.exe – исполняемый файл программы. Shipments.txt, shops.txt, warehouses.txt – текстовые файлы, которые хранят информацию сохраненных складов, магазинов, отгрузок. Кодировка данных в файле – ANSI. Программа не гарантирует корректную работу, при ручном изменении (если данные введены неверно) какого-либо из файлов: shipments.txt, warehouses.txt, shops.txt. Рекомендуется сохранять данные при помощи комбинации клавиш Ctrl+S или через вкладку программы «Файл», а затем «Сохранить». Данные в файле shipments.txt должен быть расположены в определенном формате: первая строка должна содержать количество отгрузок, последующие строки содержат информацию о каждой отгрузке в соответствующем формате. Первая строка каждой отгрузки – ее название, вторая строка – ее идентификатор, третья – идентификатор отправителя, четвертая – идентификатор получателя, пятая – название продукта, участвующего в отгрузке, шестая – количество условных единиц товара, которое участвует в отгрузке. Данные в файле shops.txt и в файле warehouses.txt должны быть также должны быть записаны в определенном формате. Первая строка файла – количество объектов (магазинов или складов), последующие строки хранят информацию о каждом объекте в определенном формате. Первая строка информации об объекте содержит название объекта, вторая – улицу, на которой расположен объект, третья – номер дома, четвертая – корпус (-1 если корпус отсутствует), пятая – вместимость объекта, шестая – занятое место, седьмая – место, зарезервированное под дальнейшие доставки, восьмая – идентификатор объекта, девятая строка – количество товаров, которые хранятся в этом объекте, далее идет описание товаров в определенном формате. Первая строка описания каждого товара - название товара, вторая – категория (пустая строка, если категория отсутствует), третья – место, занимаемое единицей товара, четвертая – количество товара, пятая – идентификатор товара, шестая – количество товара, которое отложено для дальнейших доставок.

После запуска приложения пользователь видит главное меню, которое показано на рисунке 6.9.



Рисунок 6.9 – Главное меню

Для загрузки данных из файла пользователь должен воспользоваться сочетанием клавиш Ctrl+O, либо вкладкой меню «Файл», а затем «Загрузить», изображенной на рисунке 6.10. При этом все данные, которые уже были внесены в приложение будут удалены.

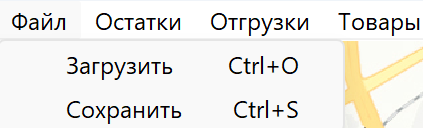


Рисунок 6.10 – Вкладка файл

После нажатия на кнопку загрузки данных появляется окно подтверждения, изображенное на рисунке 6.11.

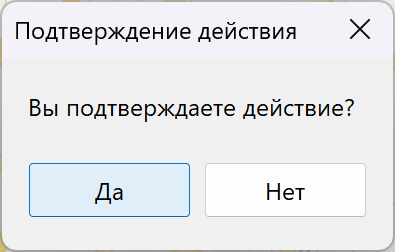


Рисунок 6.11 – Подтверждение загрузки данных

После загрузки данных пользователь увидит сообщение, изображенное на рисунке 6.12.

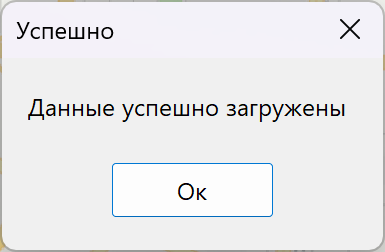


Рисунок 6.12 – Успешная загрузка данных

Для того, чтобы добавить объект на карту, пользователь должен нажать на место, где будет создан новый объект. Появится окно, показанное на рисунке 6.13.

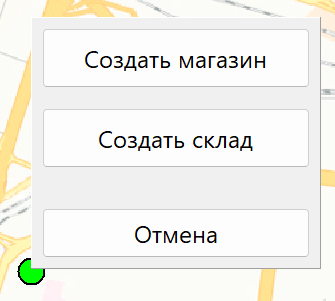


Рисунок 6.13 – Окно выбора типа объекта

После выбора типа объекта пользователь должен ввести данные для нового объекта. Окно ввода данных показано на рисунке 6.14.

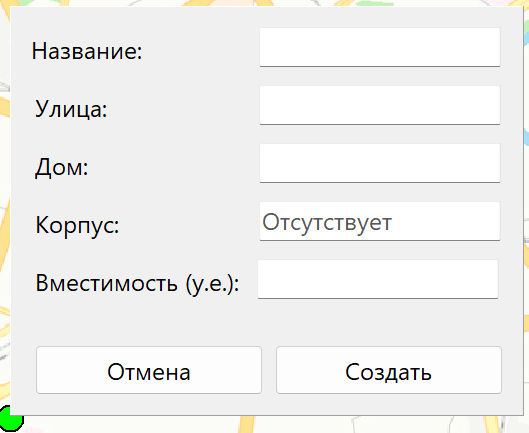


Рисунок 6.14 – Ввод данных объекта

Все данные проходят проверку при вводе пользователя. При неверном вводе пользователь увидит, какие именно данные введены неверно. Примеры представлены на рисунке 6.15.

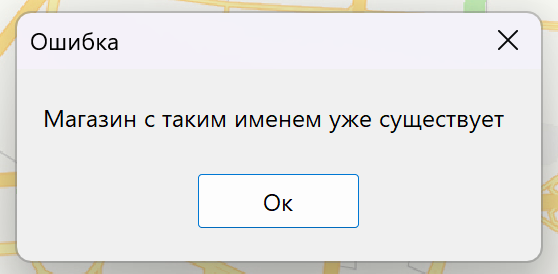
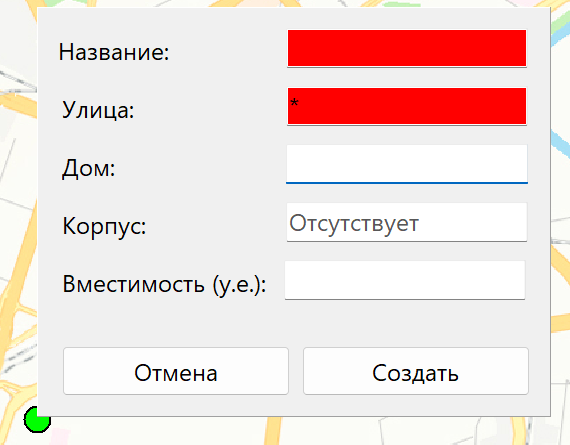


Рисунок 6.15 – Проверка данных ввода

Склады в программе отображаются бордовым цветом, магазины – синим. При наведении на объект появляется вся информация о нем. Примеры представлены на рисунке 6.16.

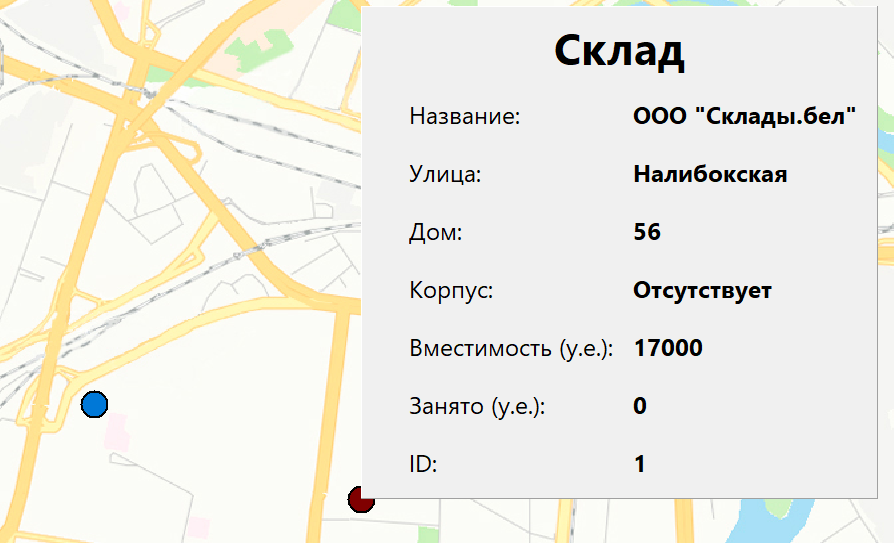


Рисунок 6.16 – Представление объектов на карте

Чтобы добавить товар в объект, на панели управления нужно нажать «Товары», затем «Добавить». Появится панель, показанная на рисунке 6.17.

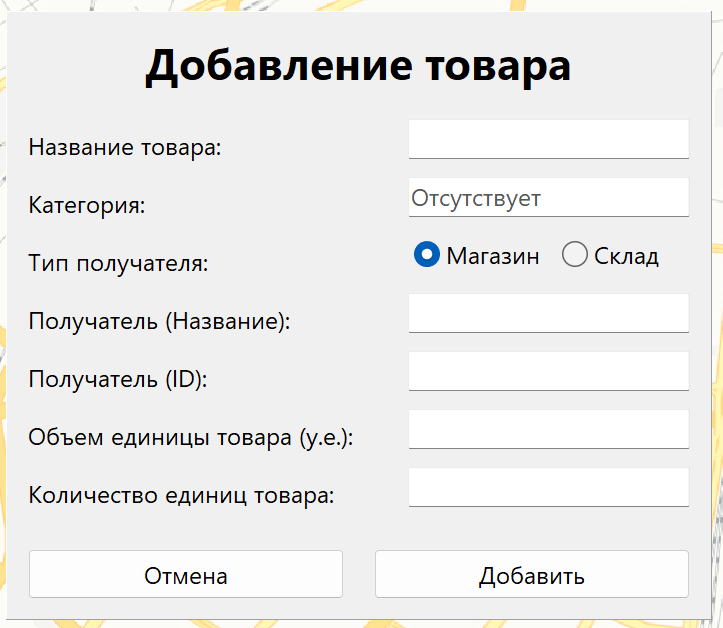


Рисунок 6.17 – Панель добавления товара

В поля панели необходимо ввести все данные товара и объекта, куда добавляется товар. Все поля проходят проверку, а в случае неверного ввода поле меняет цвет на красный, либо появляется окно ошибки, показанное на рисунке 6.15.

Для того, чтобы добавить отгрузку, необходимо нажать «Отгрузки» на панели управления, затем «Создать». Появится окно, показанное на рисунке 6.18.

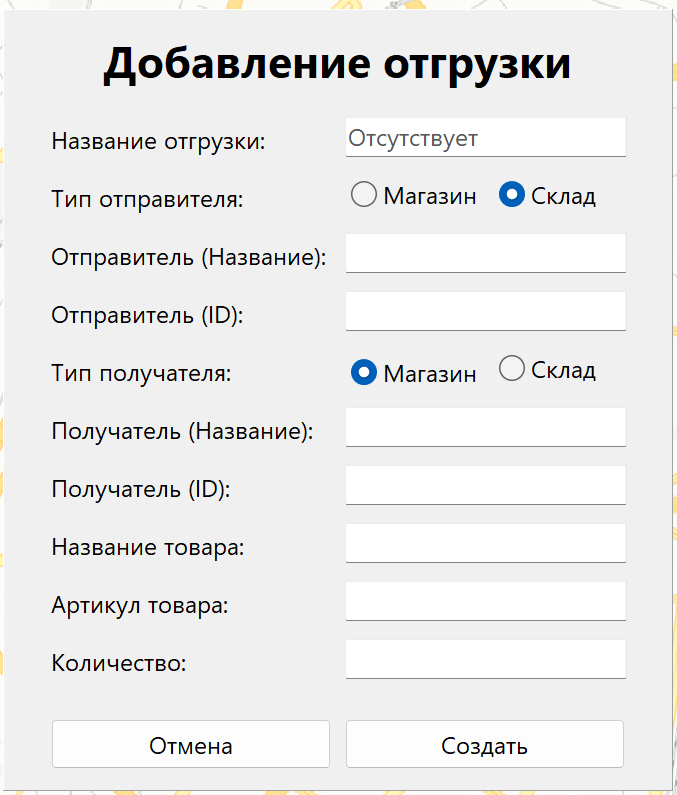


Рисунок 6.18 – Панель добавления отгрузки

Все поля проходят проверку. При неверном вводе возникает одна из ошибок, которые показаны на рисунке 6.15. Поля связаны между собой: при корректном заполнении одного поля ввода, может автоматически заполняться другое. Созданные отгрузки показаны на карте черными линиями, при наведении на которые пользователь видит информацию об отгрузке. Пример показан на рисунке 6.19.

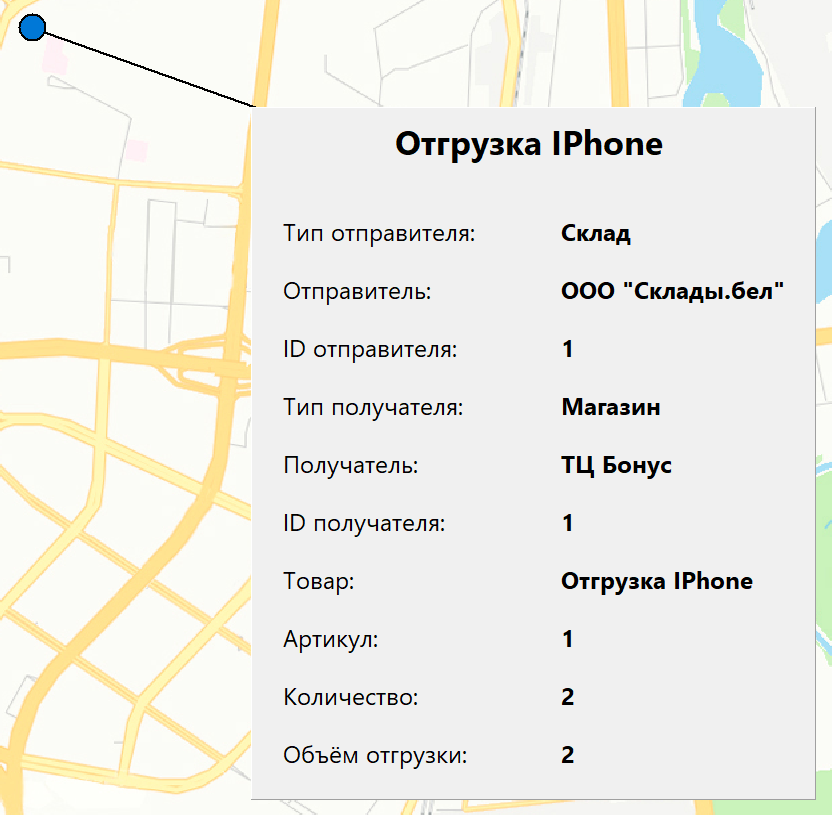


Рисунок 6.19 – Пример отгрузки

Чтобы выполнить отгрузки, можно воспользоваться пунктом меню «Отгрузки», а затем «Выполнить». Далее пользователю дается выбор между «Все», для выполнения всех существующих отгрузок и «Выбрать», если нужно выполнить какие-то конкретные отгрузки. При нажатии «Все» появляется окно с подтверждением, а затем уведомление об успешном выполнении. При нажатии «Выбрать» появляется новая форма, на которой изображена информация о существующих отгрузках, пользователь может настроить, какие именно отгрузки он хочет выполнить. Пример показан на рисунке 6.20.

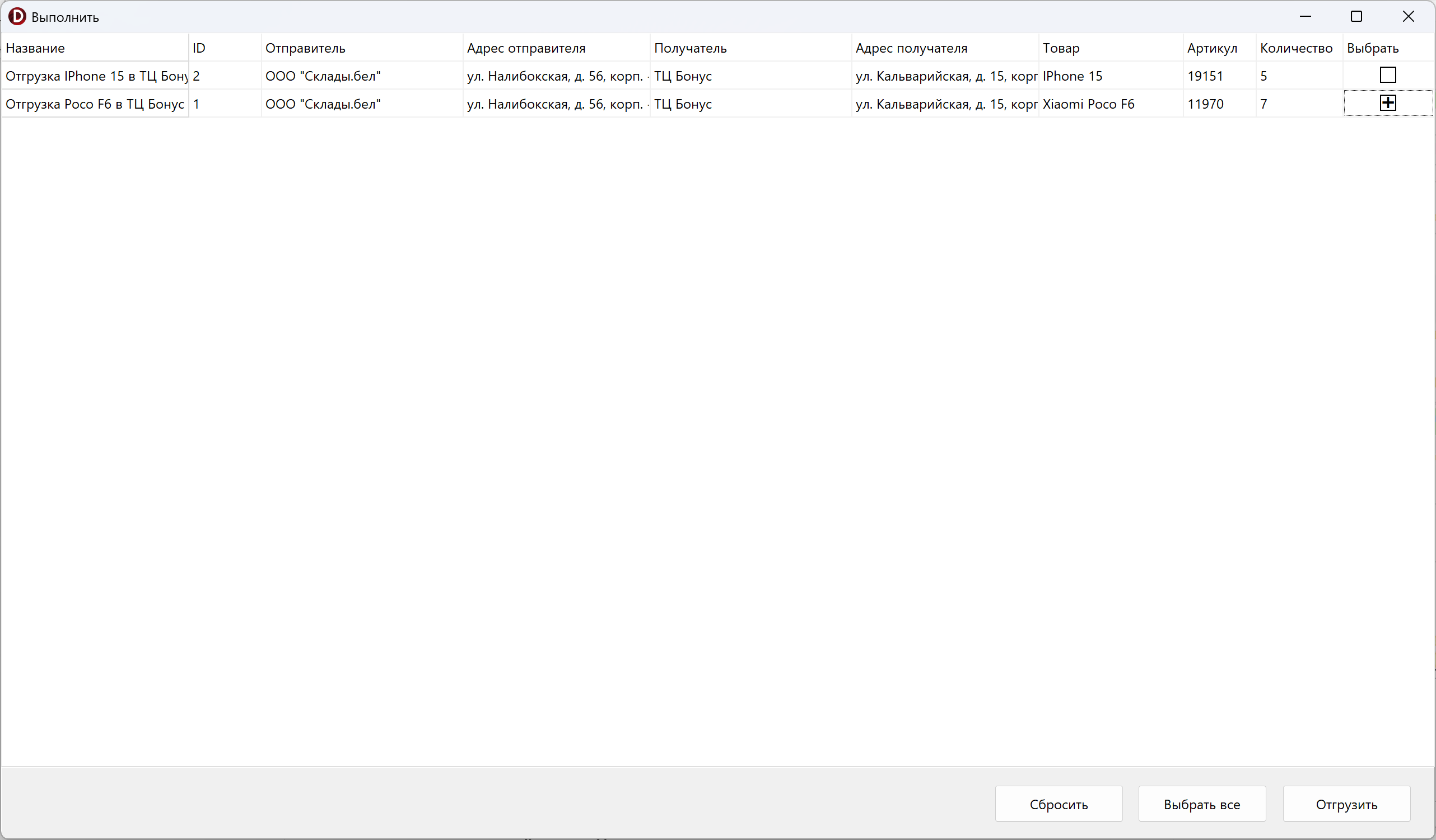


Рисунок 6.20 – Окно выполнения отгрузок

Для того, чтобы просто посмотреть список актуальных отгрузок, пользователь может воспользоваться вкладкой меню «Отгрузки», а затем «Показать». Появляется окно, показанное на рисунке 6.21.

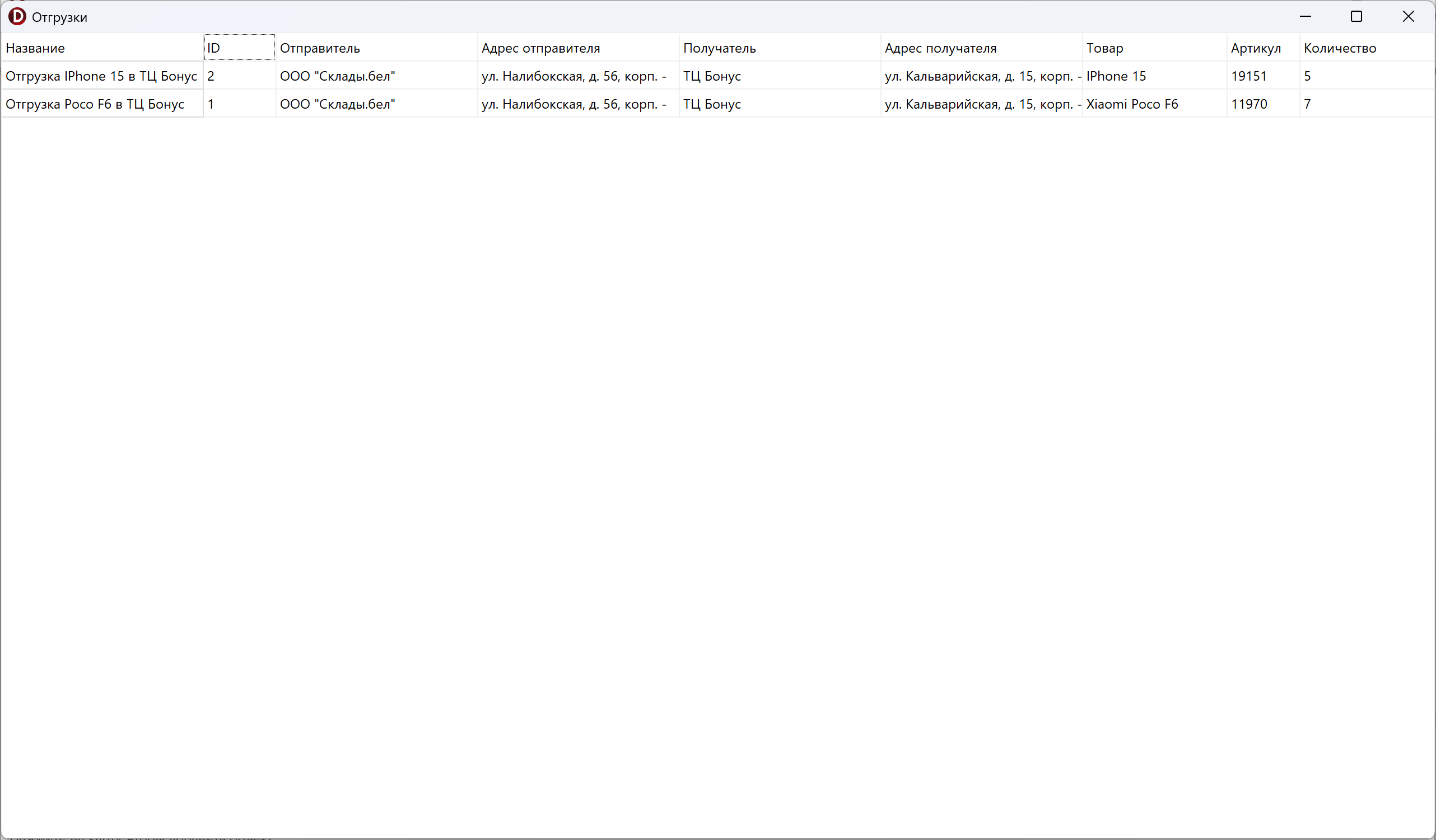


Рисунок 6.21 – Просмотр отгрузок

Для просмотра остатков в объектах, пользователь может воспользоваться вкладкой «Остатки», а затем «Показать». Появляется окно, изображенное на рисунке 6.22, в котором пользователь видит информацию о всех остатках.

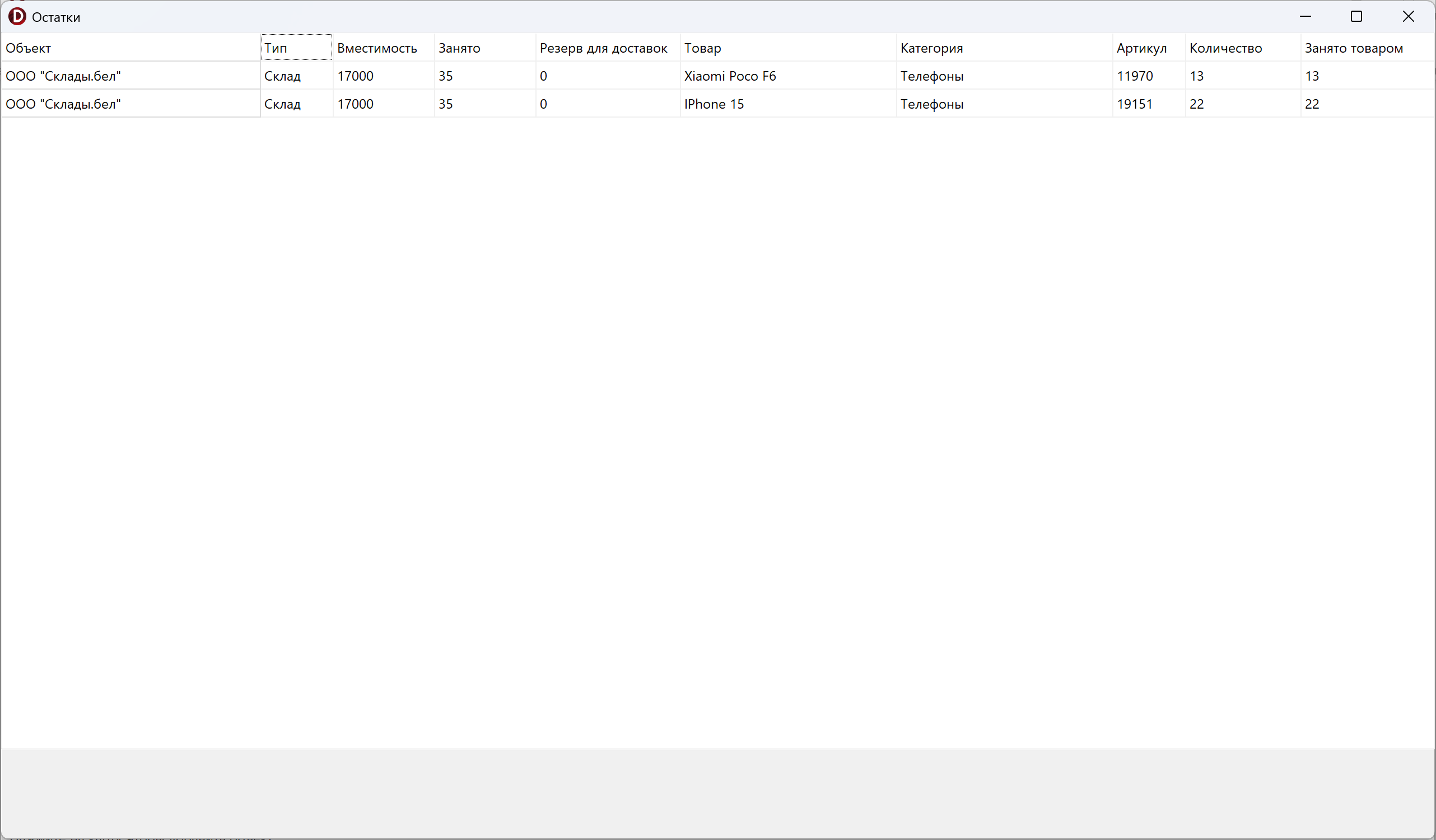


Рисунок 6.22 – Остатки в объектах

При нажатии на объект на карте появляется панель, показанная на рисунке 6.23 в которой можно посмотреть список товаров в данном объекте, удалить или изменить этот объект.

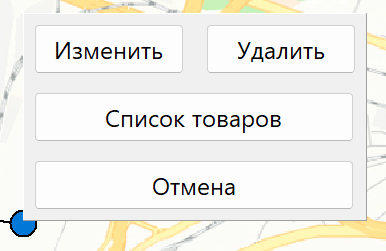


Рисунок 6.23 – Панель конкретного объекта

При нажатии кнопки «Список товаров» пользователь увидит окно, показанное на рисунке 6.24, в котором будут отображены все товары, которые в данный момент есть в этом объекте.

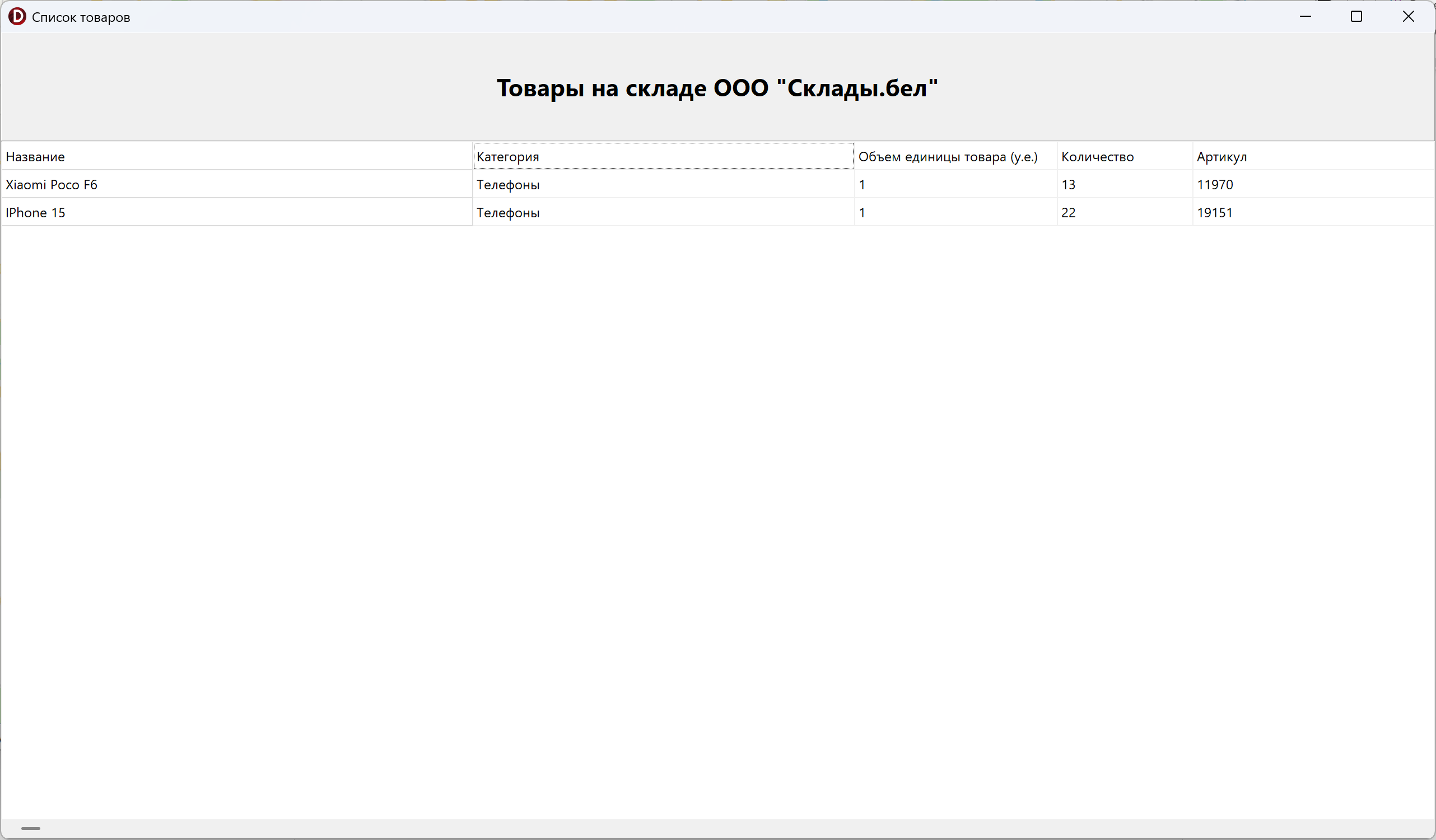


Рисунок 6.24 – Список товаров в объекте

На главном есть кнопка «Фильтр», при нажатии на которую появляется панель, показанная на рисунке 6.25.

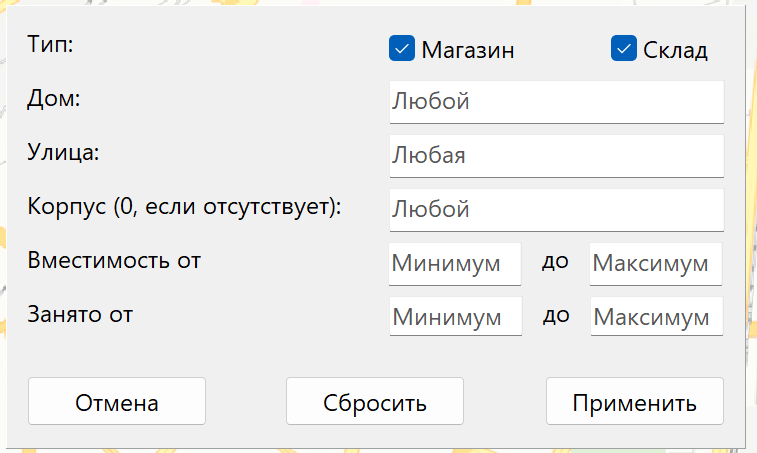


Рисунок 6.25 – Панель фильтров

При нажатии на кнопку «Сбросить» фильтр будет сброшен до исходного состояния, показанного на рисунке 6.25. Для применения фильтров пользователю необходимо ввести нужные ему значения и нажать кнопку «Применить». После применения фильтров на карте останутся только объекты, которые подходят под текущий фильтр, а количество применяемых фильтров будет отображаться рядом с кнопкой «Фильтр» как показано на рисунке 6.26

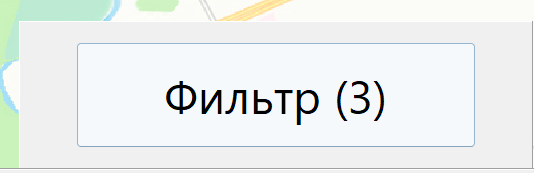


Рисунок 6.26 – Количество применяемых фильтров

Закрыть программу можно при помощи крестика, который расположен в правом верхнем углу окна и показан на рисунке 6.27.



Рисунок 6.27 – Кнопка закрытия программы

Если есть данные, которые не были сохранены, пользователь увидит окно подтверждения, которое показано на рисунке 6.28.

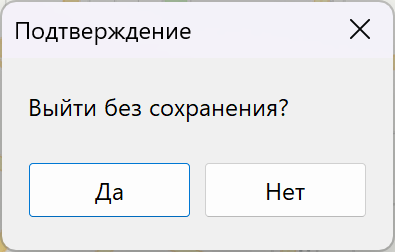


Рисунок 6.28 – Подтверждение выхода