МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт информационных технологий и управления в технических системах

Кафедра информационных технологий и компьютерных систем

## Пояснительная записка

к курсовой работе по дисциплине

«Программирование. Языки высокого уровня»

***Разработка алгоритма и программы типового вычислительного***

на тему\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***процесса***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***ИВТ/б-19-2- о***

***2***

Выполнил: студент \_\_\_\_\_\_\_ курса, группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

направления подготовки 09.03.01 — Информатика и вычислительная техника

***Садриев АланЭмильевич***

***Волкова Т.В., к.т.н., доцент,***

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, степень, звание, должность)

***доцент кафедры ИТКС***

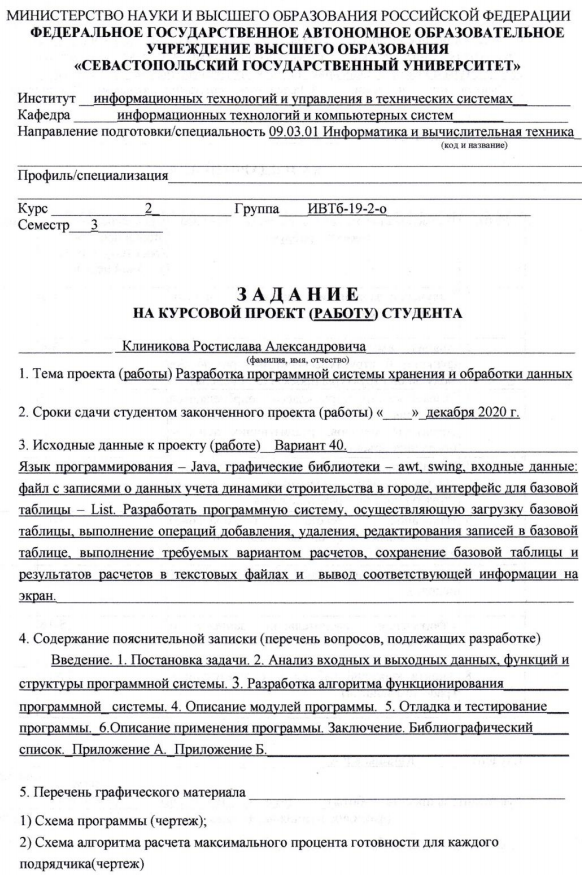
Дата допуска к защите «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

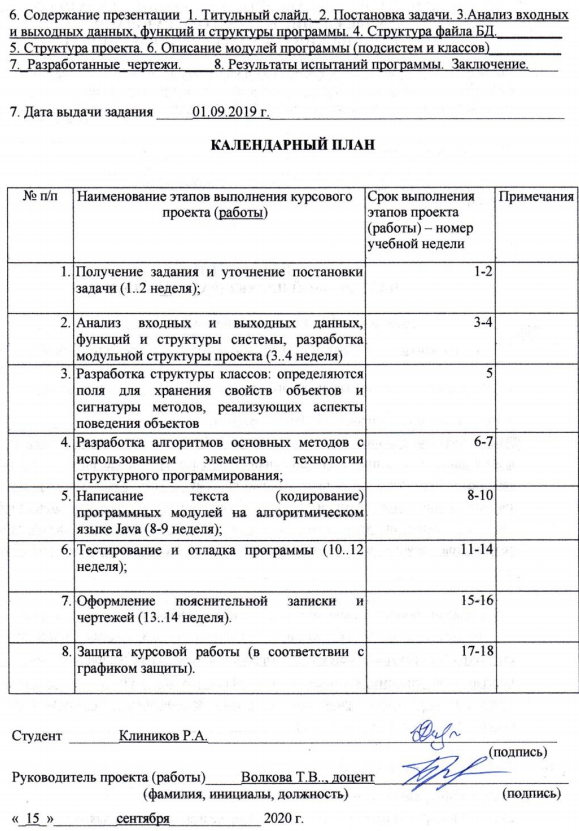
***А.А Брюховецкий***

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (инициалы, фамилия)

2020 г.





**СОДЕРЖАНИЕ**

[Пояснительная записка 4](#_Toc58362916)

[ВВЕДЕНИЕ 8](#_Toc58362917)

[1. ПОСТАНОВКА ЗАДАЧИ 9](#_Toc58362918)

[2. АНАЛИЗ ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ, ФУНКЦИЙ И СТРУКТУРЫ ПРОГРАММЫ 11](#_Toc58362919)

[3. РАЗРАБОТКА АЛГОРИТМА ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОЙ СИСТЕМЫ 13](#_Toc58362920)

[4. ОПИСАНИЕ МОДУЛЕЙ ПРОГРАММЫ 14](#_Toc58362921)

[5. ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ 27](#_Toc58362922)

[6. ОПИСАНИЕ ПРИМЕНЕНИЯ ПРОГРАММЫ 38](#_Toc58362923)

[ЗАКЛЮЧЕНИЕ 39](#_Toc58362924)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 40](#_Toc58362925)

[ПРИЛОЖЕНИЕ А 41](#_Toc58362926)

# **In dev ВВЕДЕНИЕ**

Согласно заданию, на курсовой проект необходимо разработать программную систему хранения и обработки данных. Актуальность темы обусловлена тем, что в современном мире неисчислимое множество информации, хранящейся в цифровом виде, и всю информацию необходимо хранить и обрабатывать. Так же в разрабатываемой программной системе обязан присутствовать графический интерфейс пользователя. Для написания программы используется язык программирования Java.

В процессе выполнения курсового проекта необходимо выполнить два чертежа:

- схема программы – чертеж «Схема программы»;

- схема алгоритма подсчета общего количества побед для каждой секции;

Интерфейс для базовой таблицы – ObservableList наследующий базовый интерфейс List.

Java Collection Framework — иерархия интерфейсов и их реализаций, которая является частью JDK и позволяет разработчику пользоваться большим количесвом структур данных из «коробки».

Java Collections Framework состоит из следующих компонентов:

List это упорядоченный список объектов (иногда еще называют последовательностью объектов). Элементы списка могут вставляться или извлекаться по их индексу в списке (индексация начинается с 0). В эту группу входят: ArrayList, LinkedList, Vector, Stack.

# **1. ПОСТАНОВКА ЗАДАЧИ**

Согласно варианту задания, необходимо разработать программную систему хранения и обработки данных на языке программирования Java. Таблица содержит данные о маршрутах городского транспорта. Один и тот же объект могут осваивать несколько подрядчиков (выполняя на нем те или иные работы в определенное время). Один и тот же подрядчик может выполнять работы на нескольких объектах.

Структура записи: код маршрута (ключевое поле), название маршрута, вид транспорта, протяженность в км, число остановок.

Подсчитать:

1) минимальное и максимальное число остановок для каждого вида транспорта.

2) общее количество маршрутов.

Условие удаления (дополнительное): протяженность ниже заданной.

Сортировка по возрастанию в поле вид транспорта, внутри одного вида – по убыванию протяженности.

Фильтрация по полю вид транспорта.

Интерфейс для базовой таблицы: ObservableList.

Также, разрабатываемая программа должна использовать графический интерфейс, обеспечивающий выполнение следующего минимального состава действий:

- создание новой таблицы («Создать»);

- загрузка таблицы из текстового файла («Открыть»);

- просмотр таблицы. При этом необходимо предусмотреть возможность скроллинга;

- добавление новой записи в таблицу;

- удаление записи по значению ключевого поля и удаление совокупности записей, удовлетворяющих заданному вариантом условию;

- редактирование (обновление) записи в таблице. Корректируемую запись выбирают по значению ключевого поля (полей);

- сортировка таблицы;

- фильтрация данных по значению в заданном поле;

- выполнение итоговых расчетов для таблицы и просмотр их результатов;

- сохранение таблицы в том же текстовом файле («Сохранить») или в файле, имя которого вводит пользователь («Сохранить как»);

- выход – завершение работы программы.

Для создания графической визуализации были использованы библиотеки JavaFX.

Для правильной и простой сборки проекта использован сборщик Gradle.

В качестве среды разработки для написания кода проекта было выбрано программное обеспечение IntelliJ IDEA.

# **2. АНАЛИЗ ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ, ФУНКЦИЙ И СТРУКТУРЫ ПРОГРАММЫ**

Объектом проектирования является программная система хранения и обработки данных. На первом этапе проектирования программная система представляется как «черный ящик» (Рисунок 2.1), детализируются ее входные и выходные данные, управляющие воздействия и выполняемые функции.



Рисунок 2.1–Представление программной системы в виде «черного ящика» с указанием входных и выходных данных и управляющих воздействий (событий)

Согласно заданию на курсовую работу, система должна выполнять следующие функции:

- создание новой базы данных;

- загрузка существующей базы данных из текстового файла;

- редактирование данных;

- выборка данных из базы данных по условиям;

- фильтрация данных при выборке;

- сортировка данных при выборке;

- выполнение итоговых вычислений над данными;

- сохранение информации в том же файле;

- сохранение информации в другом файле.

В результате анализа функций, систему можно разбить на отдельные, связанные между собой подсистемы. Итоговая структура системы изображена на рисунке 2.2.



Рисунок 2.2– Структура программной системы

В данной структуре подсистема графического интерфейса пользователя, обращаясь к подсистеме ввода-вывода, инициирует ввод строк из текстового файла в список строк, затем, с помощью подсистемы перевода строк в записи обратно, преобразует список строк в список записей базы данных. Пример файла исходных данных приведен на рисунке 2.3.

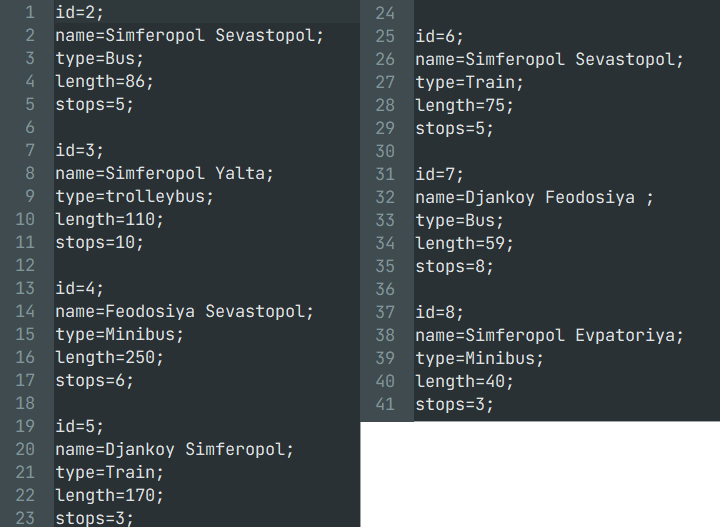


Рисунок 2.3– Пример файла базы данных

# **3. РАЗРАБОТКА АЛГОРИТМА ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОЙ СИСТЕМЫ**

# IN DEV

На чертеже «СевГУ.2020.ИВТб-19-2-о.34.01» видно, что программа состоит из 5 частей:

- подсистема запуска ГИП. Выполняет только функцию запуска программы;

- подсистема ГИП: данный блок содержит в себе все визуальные компоненты, с которыми будет работать пользователь;

- подсистема ввода-вывода: выполняет функцию ввода записей в базу данных из текстового файла и наоборот;

- подсистема перевода строк в записи и обратно: выполняет преобразование строк из текстового файла в запись базы данных и обратно;

- подсистема обработки внутреннего представления базы данных: реализует форму представления базы данных, в данной подсистемы выполняются все алгоритмы обработки и редактирования базы данных.

На чертеже «СевГУ.2020.ИВТб-19-2-о.34.02» показан алгоритм функционирования разрабатываемой системы.

На чертеже «СевГУ.2020.ИВТб-19-2-о.34.03» детализирован блок выполнения пункта меню «Открыть».

На чертеже «СевГУ.2020.ИВТб-19-2-о.34.04» показан алгоритм удаления группы записей.

Чертежи алгоритмов функционирования программы представлены в приложении Б.

# **4. ОПИСАНИЕ МОДУЛЕЙ ПРОГРАММЫ**

Подсистема запуска реализуется классом Main, который в свою очередь запускает класс Run, запускающий главную подсистему графической оболочки.

Подсистема ввода-вывода и преобразования текстовой информации в список реализуется методом “fromFileToList”, расположенным в классе Global, преобразование списка в файл осуществляется в методе “fromListToFile”, который также располагается в классе Global.

Подсистема обработки внутреннего представления БД (так называемы Back-End программы) реализуется совокупностью классов:

Route – описывает свойства и поведение объекта-записи БД.

Routes – описывает свойства и поведение группы объектов класса

Route (внутреннее представление таблицы БД).

В подсистему ГИП (или же Front-End программы) входят следующие классы:

Indev AboutWindow

MainWindow – описывает главное окно системы;

Header – описывает меню взаимодействия пользователя и программы;

ModalWindow – описывает работу всплывающих окон;

OnCloseRequest – описывает работу всплывающего окна при закрытии;

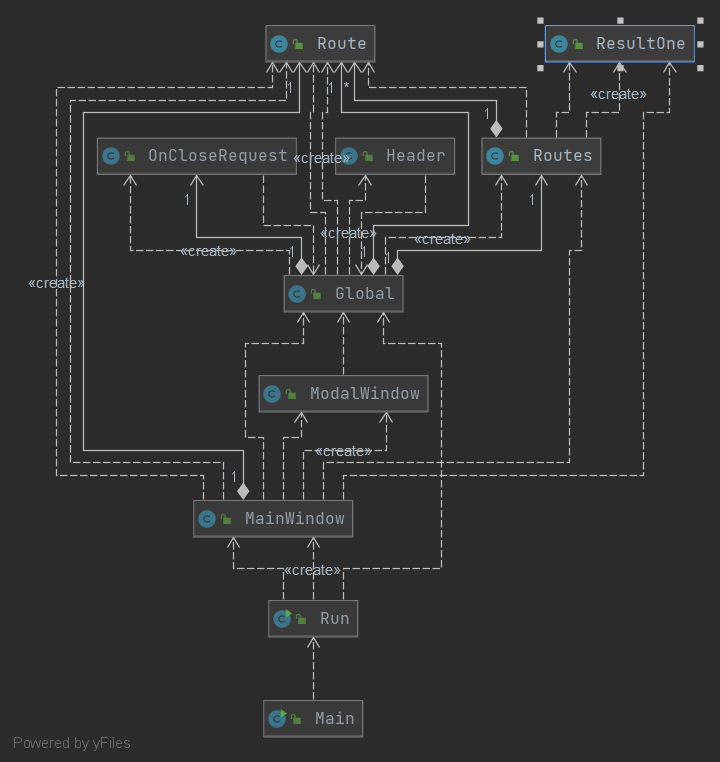


Рисунок 4.1– Структура проекта, реализующего работу с базой данных «Маршруты»

Как видно из рисунка 4.1, подсистема ГИП и подсистема обработки внутреннего представления базы данных взаимодействуют посредством класса Global, в котором определены глобальные public-поля и методы проекта.

Далее будут описаны все поля и методы разработанных классов.

Класс Main:



Рисунок 4.2 – Структура класса Main

Методы:

* public static void main (String [] args) – стартовая точка программы

Класс Run:

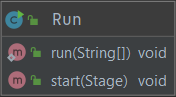


Рисунок 4.3 – Структура класса Run

Методы:

* public static void run (String [] args) – статический метод наследуемый из класса Application, запускающий разработанное приложение.
* public void start(Stage primaryStage) – абстрактный метод класса Application, который осуществляет инициализацию формы GUI.

Класс Global:

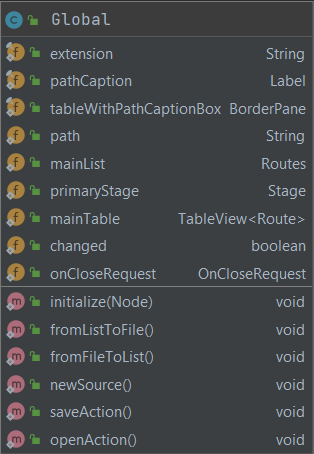


Рисунок 4.4 – Структура класса Global

Поля:

* public static String extention – раширение сохраняемого файла
* public static Label pathCaption – метка для отображения полного пути до файла в программе
* public static BorderPane tableWithPathCaptionBox – панель компоновки для отображения полного пути к файлу и основной таблицы
* public static String path – полный путь к файлу базы данных.
* public static Routes mainList – главный список хранящий записи из файла
* public static Stage primaryStage – главное окно программы
* public static TableView<Route> mainTable – основная таблица отображаемая в программе
* public static boolean changed – поля для отслеживание наличия изменений в таблице.
* public static OnCloseRequest onCloseRequest – объект отображения подтверждения при выходе из программы

Методы:

* public static void initialize(Node node) – инициализация главной сцены
* public static void fromListToFile() – преобразование и вывод основного списка в файл.
* public static void fromFileToList() – преобразование и ввод файла в основной список программы
* public static void newSource() – создание нового (чистого) списка
* public static void saveAction() – метод сохранения файла
* public static void openAction() – метод отрытия файла
* private static String savedFileName() – метод создания имени файла в формате “database\_yyyy-MM-dd\_HH-mm-ss”

Класс Route:

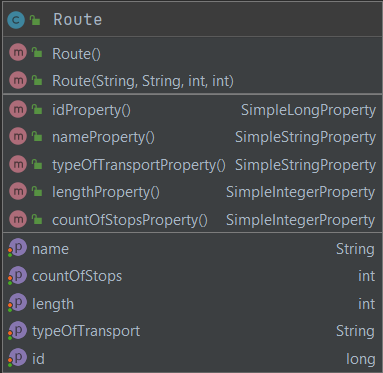


Рисунок 4.5 – Структура класса Route

Поля:

* private String name – название маршрута
* private int countOfStops – количество остановок
* private int length – протяженость маршрута
* private String typeOfTransport – тип транспорта
* private long id – код маршрута

Методы:

* public Route() – конструктор без параметров.
* public Route(String name, String typeOfTransport, int length, int countOfStops) – конструктор с параметрами.
* И далее идут методы возвращающие и устанавливающие значения в private поля класса.

Класс Routes:

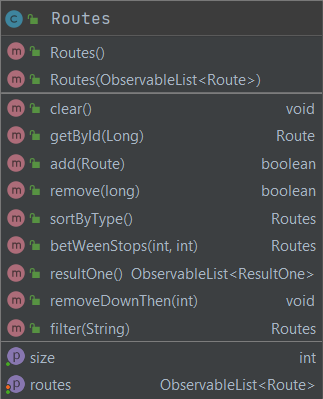


Рисунок 4.6 – Структура класса Routes

Поля:

* private ObservableList<Route> routes – (отслеживаемый) список объектов
* private Long count – начальный идентификационный номер объекта

Методы:

* public Routes() – конструктор без параметров
* public Routes(ObservableList<Route> routes) – конструктор с параметрами
* public void clear() – метод очищения таблицы
* public ObservableList<Route> getRoutes() – метод возвращающий текущий список
* public void setRoutes() – метод устанавливающий значения текущему списку
* public Route getById(Long id) – метод возвращающий объект по заданному идентификационному номеру
* public boolean add(Route route) – метод добавления объекта в текущий список
* public boolean remove(long id) – метод удаления объекта из текущего списка по заданномк идентификационному номеру
* public Routes sortByType() – метод сортировки текущего списка по заданному условию в задании
* IN DEV public Routes betweenStops(int min, int max) – метод
* public ObservableList<ResultOne> resultOne – метод реализующий итог 1
* public int resultTwo() – метод реализующий итог 2
* public Routes filter(String type) – метод реализующий фильтрацию по текущему списку

Класс ResultOne:

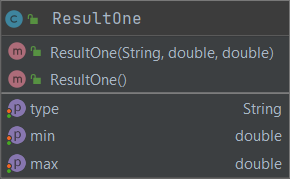


Рисунок 4.7 – Структура класса ResultOne

Поля:

* private String type – поля для хранения типа транспорта
* private double min – поле для хранения минимального количества остановок для данного вида транспорта
* private double max – поле для хранения максимального количества остановок для данного вида транспорта

Методы:

* public ResultOne() – конструктор без параметров
* public ResultOne(String type, double min, double max) – конструктор с параметрами
* Далее реализуются методы возвращающие и устанавливающие иначения для private полей класса

Класс Header:

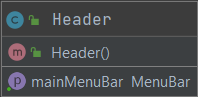


Рисунок 4.7 – Структура класса Header

Поля:

* private final MenuBar mainMenuBar – объект содержащий элементы меню
* В блоке кода объявления полей, создаются объекты для создания и отображения элементов меню.

Методы:

* public Header() – конструктор без параметров, в котором инициализируется элементы меню.
* public MenuBar getMainMenuBar – метод возвращающий объект с элементами меню

Класс ModalWindow:

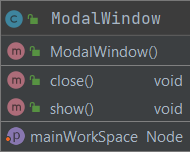


Рисунок 4.8 – Структура класса ModalWindow

Поля:

* private final Stage STAGE – создание рабочей области всплывающего окна

Методы:

* public ModalWindow() – конструктор без параметров, в котором инициализируется главная сцена окна
* public void setMainWorkSpace(Node node) – метод добавляющий в главную сцену пользовательские элементы
* public void close() – метод закрывающий модальное окно
* public void show() – метод запускающий модальное окно

Класс OnCloseRequest:

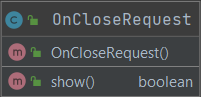


Рисунок 4.9 – Структура класса OnCloseRequest

Поля:

* private final Alert alert – поле основного всплывающего окна с предупреждением

Методы:

* public OnCloseRequest() – конструктор без параметров, инициализирующий логику работы данного класса
* public boolean show() – метод запускающий модальное предупреждающее окно

Класс MainWindow:

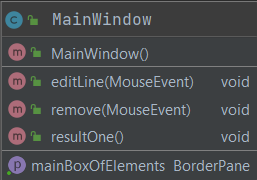


Рисунок 4.11 – Структура класса MainWindow

Поля:

* private final StackPane tableRoot – компоновщик для элемента таблица
* private final BorderPane mainBoxOfElements – главный компоновщик всех элементов окна
* Далее создаются поля класса TableColumn<>, являющиеся столбцами основной таблицы программы

Методы:

* public MainWindow() – конструктор без параметров, в котором инициализируется вся логика создания и управления таблицей
* public BorderPane getMainBoxOfElements – метод возвращающий ссылку на главный компоновщик всех элементов окна
* public void addToTable(Routes routes) – метод добавляющий в таблицу элементы и переданного в него класса “Routes”
* public void editLine(MouseEvent mouseEvent) – метод реализующий редактирование таблицы
* public void remove(MouseEvent mouseEvent) – метод реализующий удаление одного элемента из таблицы
* public void resultOne() – метод реализующий логику вывода Итога 1

Место ОСТАНОВКИ

# **5. ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ**

Главное меню системы показано на рисунке 5.1. Вид главного окна при начале работы системы изображен на рисунке 5.2.

С помощью вкладки «Редактирование», расположенной в верхней части программы, пользователь может взаимодействовать (добавлять, удалять, редактировать) информацию в базу данных. Для добавления записи необходимо ввести информацию в четыре текстовых поля (для ввода кода подрядчика, названия объекта, даты, процента готовности) и нажать кнопку «Добавить». Для удаления записей из таблицы нужно заполнить три текстовых поля и нажать кнопку «Удалить». С помощью кнопки «Удалить группу записей» можно удалить записи процент готовности ниже заданного. Кнопка «Обновить» позволить редактировать дату и процент готовности у записи. Для этого необходимо ввести ключ, который составляют первые три поля. Затем следует ввести дату и процент готовности, на которые нужно заменить старые значения.

На рисунке 5.3 проиллюстрировано выполнение команды главного меню File → Open.

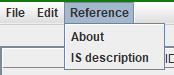
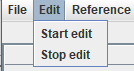
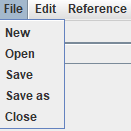


Рисунок 5.1– Главное меню программы

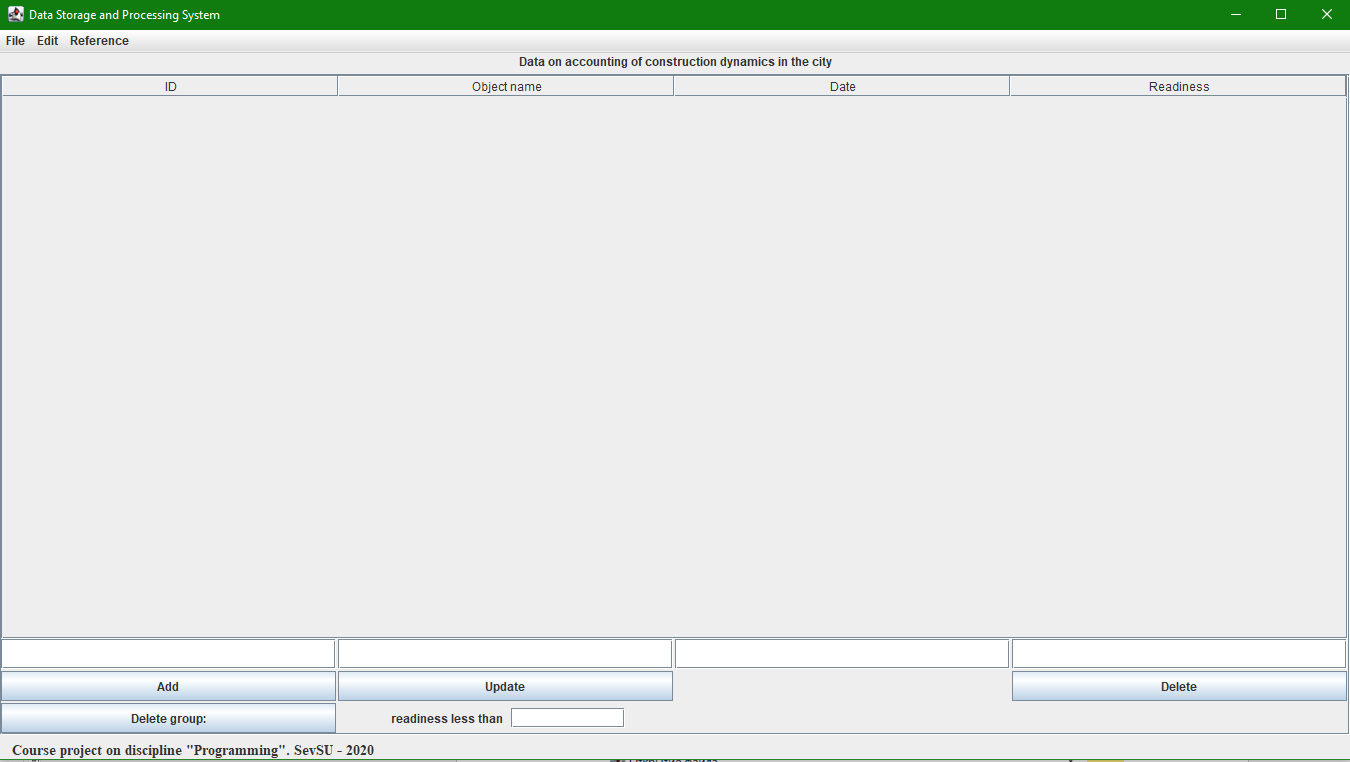


Рисунок 5.2– Вид главного окна при начале работы системы

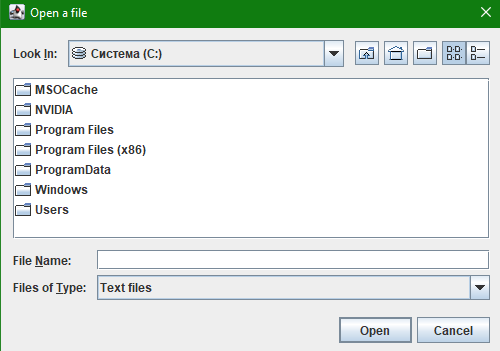


Рисунок 5.3– Вид окна диалога MenuIS после выбора в меню команды File→Open

После выполнения команды File→Open в главном окне отображаются в виде таблицы данные, введенные из выбранного файла. Выполнение запросов на выборку данных и подведения итоговых расчетов в результате нажатия кнопок «Max readiness for every builder company» и «Count of objects» (Рисунок5.5 – 5.8).

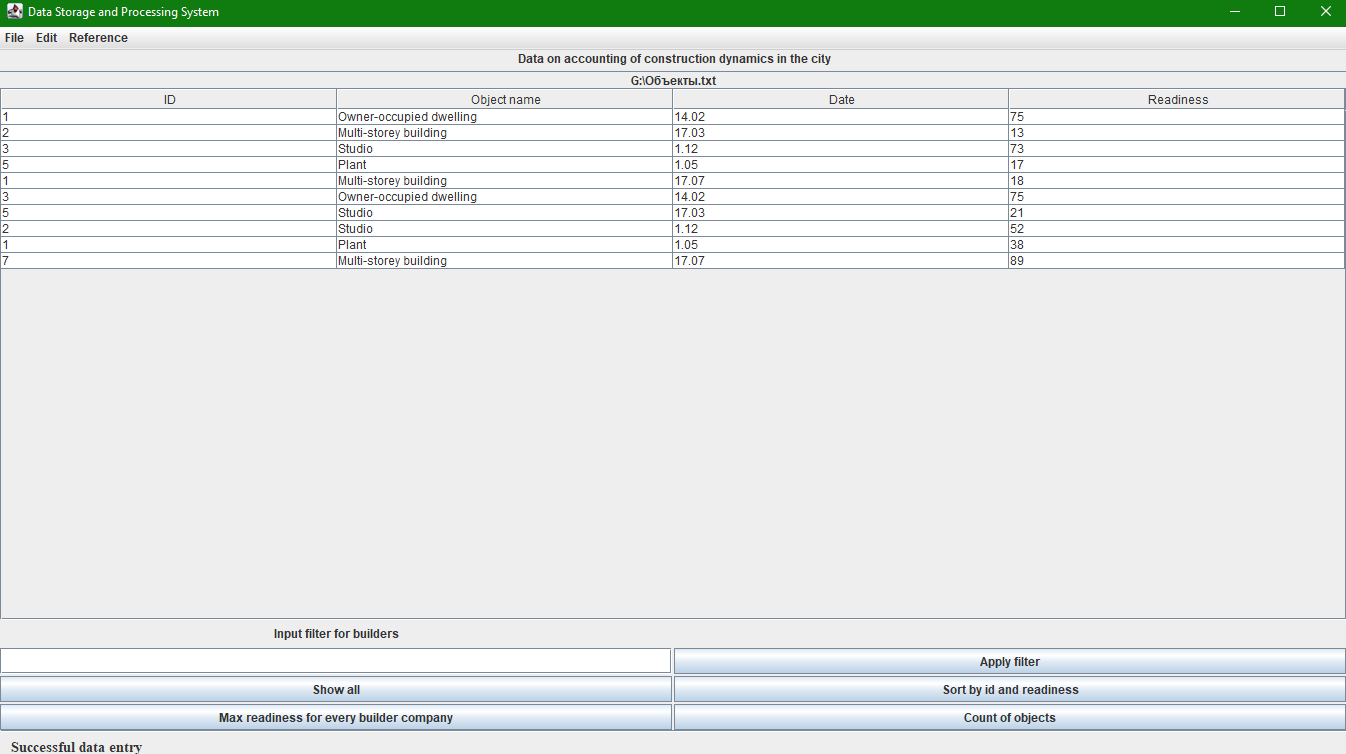


Рисунок 5.5– Вид главного окна после выполнения команды File→Open

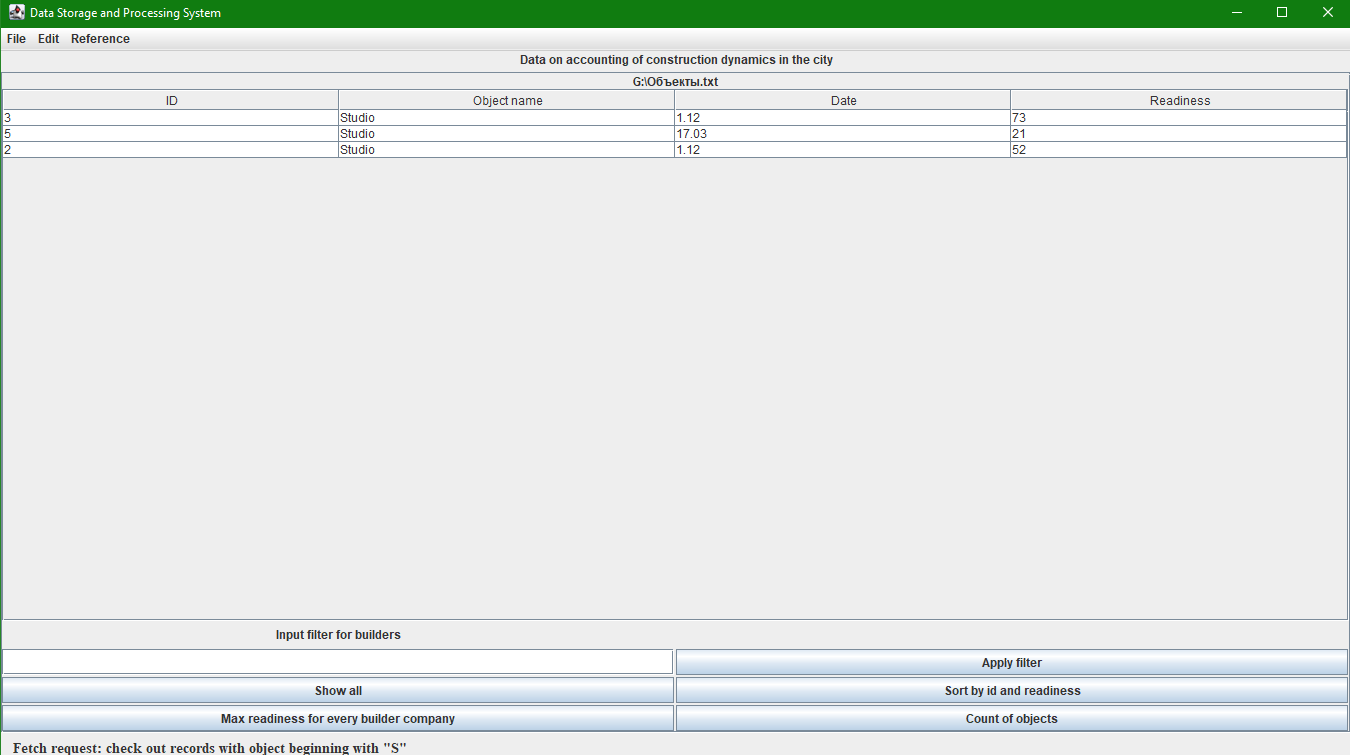


Рисунок 5.6– Вид главного окна после ввода в текстовое поле фильтрация

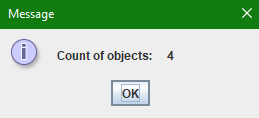


Рисунок 5.7 – Окно итогового расчета «Общее число объектов»

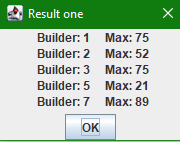


Рисунок 5.8 – Окно итогового расчета «Максимальный процент готовности для каждого подрядчика»

Таблица после нажатия кнопки «Show all» (Рисунок 5.9).

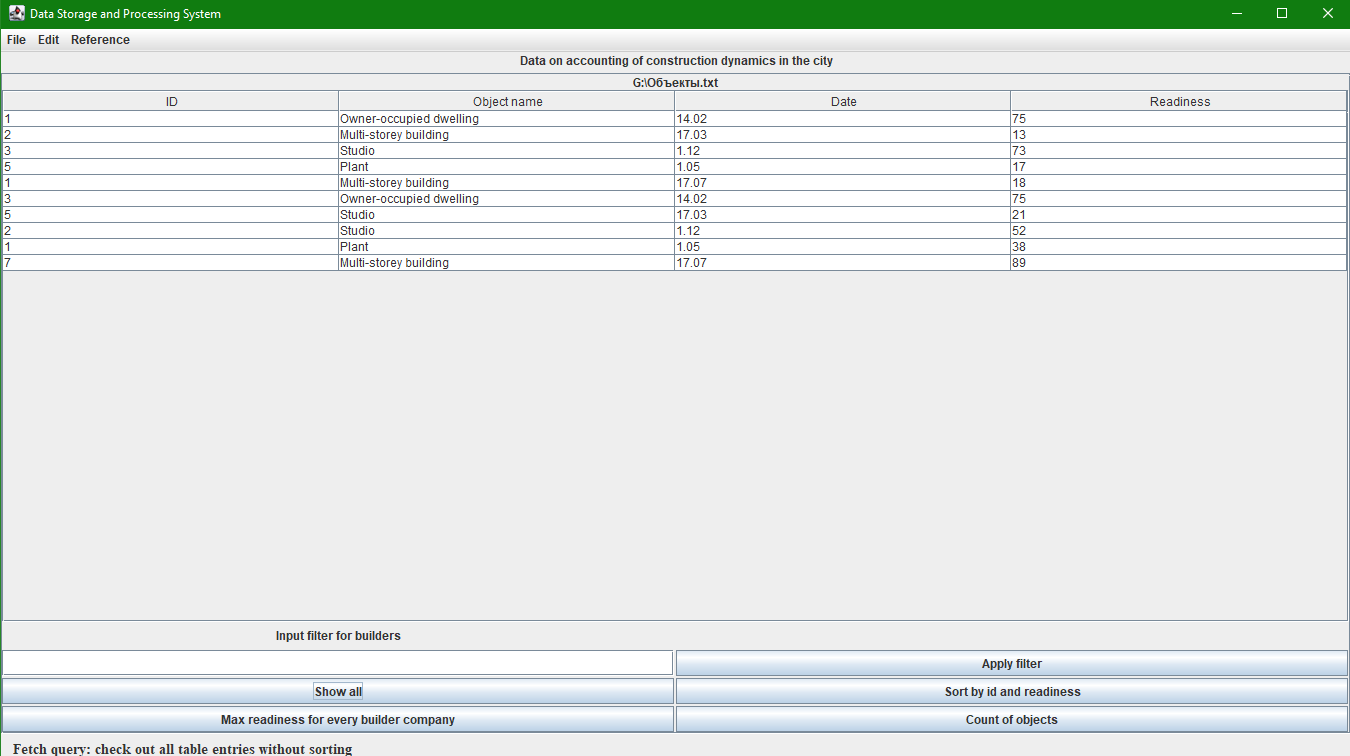


Рисунок 5.9 – Вид главного окна после нажатия кнопки «Show all»

Отсортировать таблицу можно нажатием кнопки «Sort by id and readiness». (Рисунок 5.10).

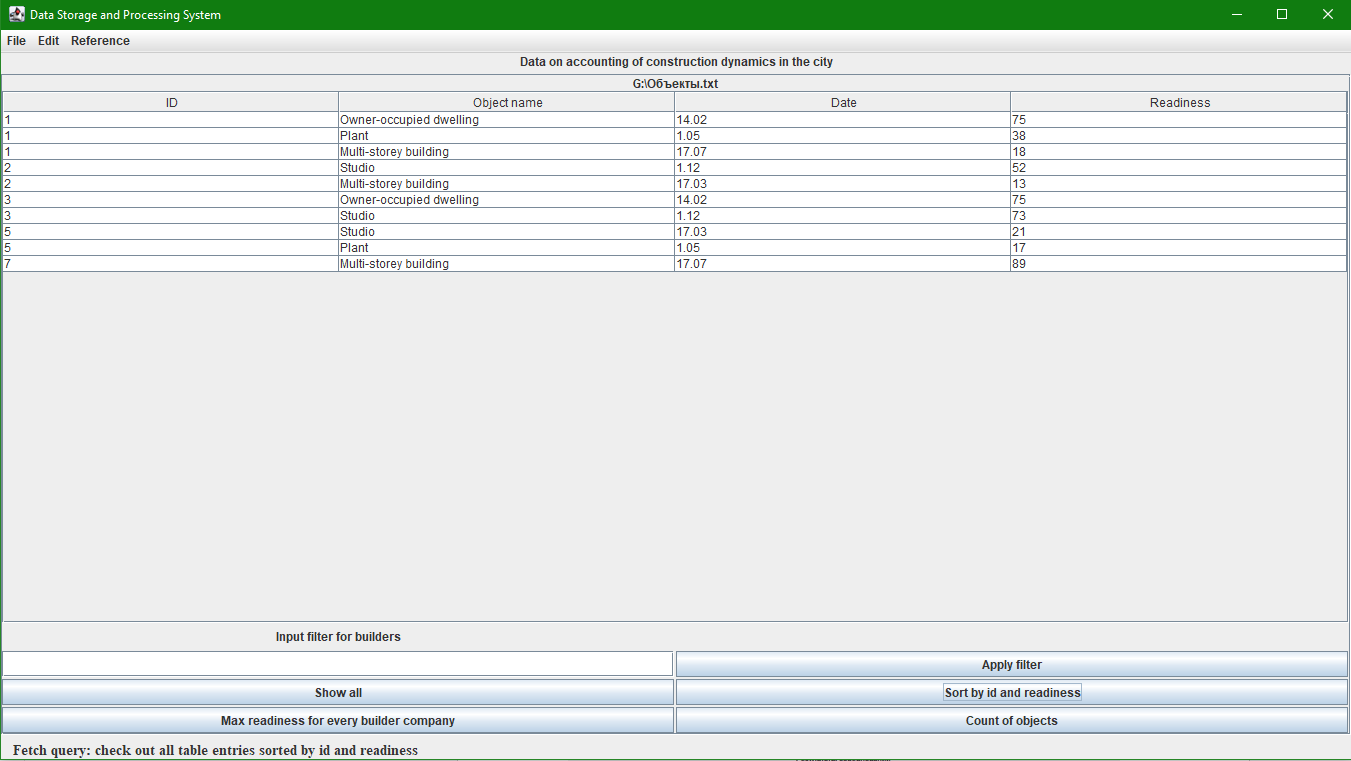


Рисунок 5.10 – Вид главного окна после нажатия кнопки «Sort by id and readiness»

Для редактирования таблицы, необходимо выполнить команду «Edit→Start edit» (Рисунок 5.12). Далее показано как нужно добавлять записи в таблицу (Рисунок 5.13), удалять запись из таблицы (Рисунок 5.14), обновлять данные в записи (Рисунок 5.15), удаление группы записей (Рисунок 5.16).

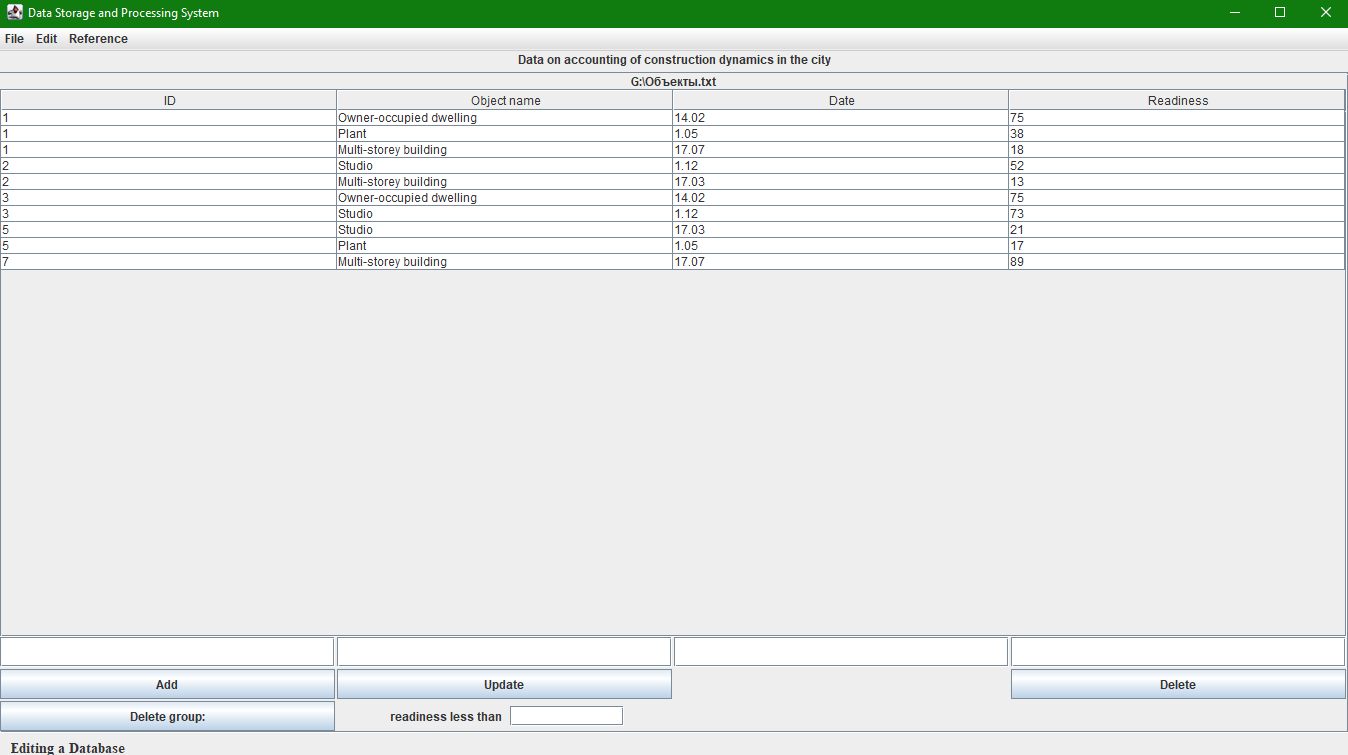
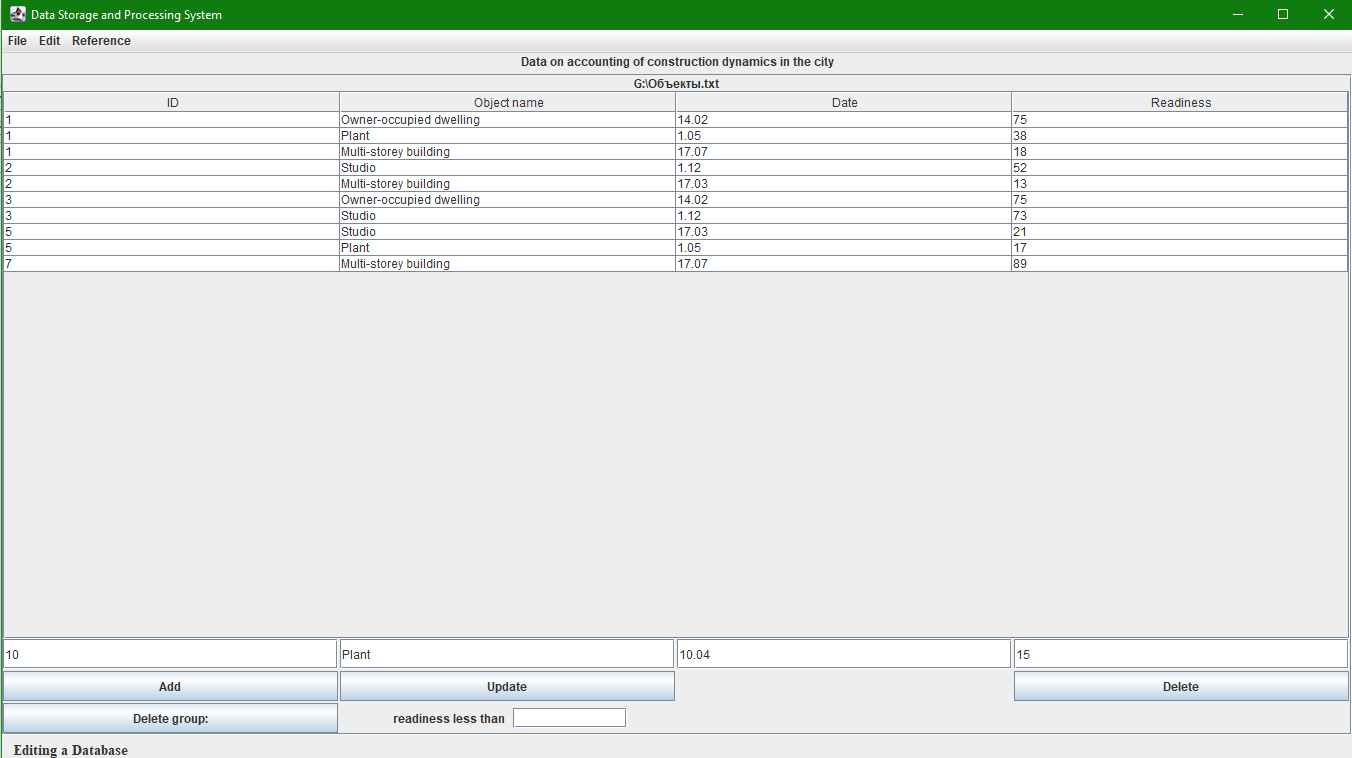


Рисунок 5.12 – Окно программы, после выполнения команды «Edit→Start edit»



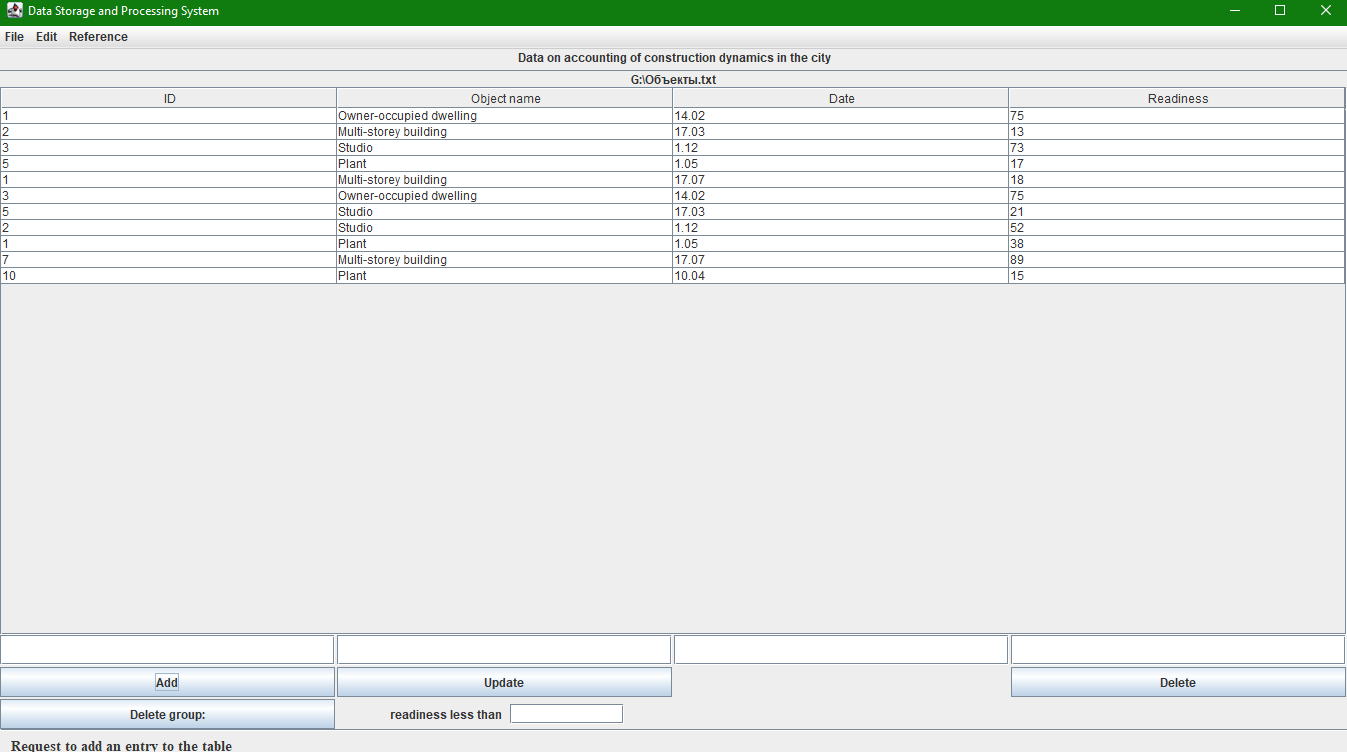
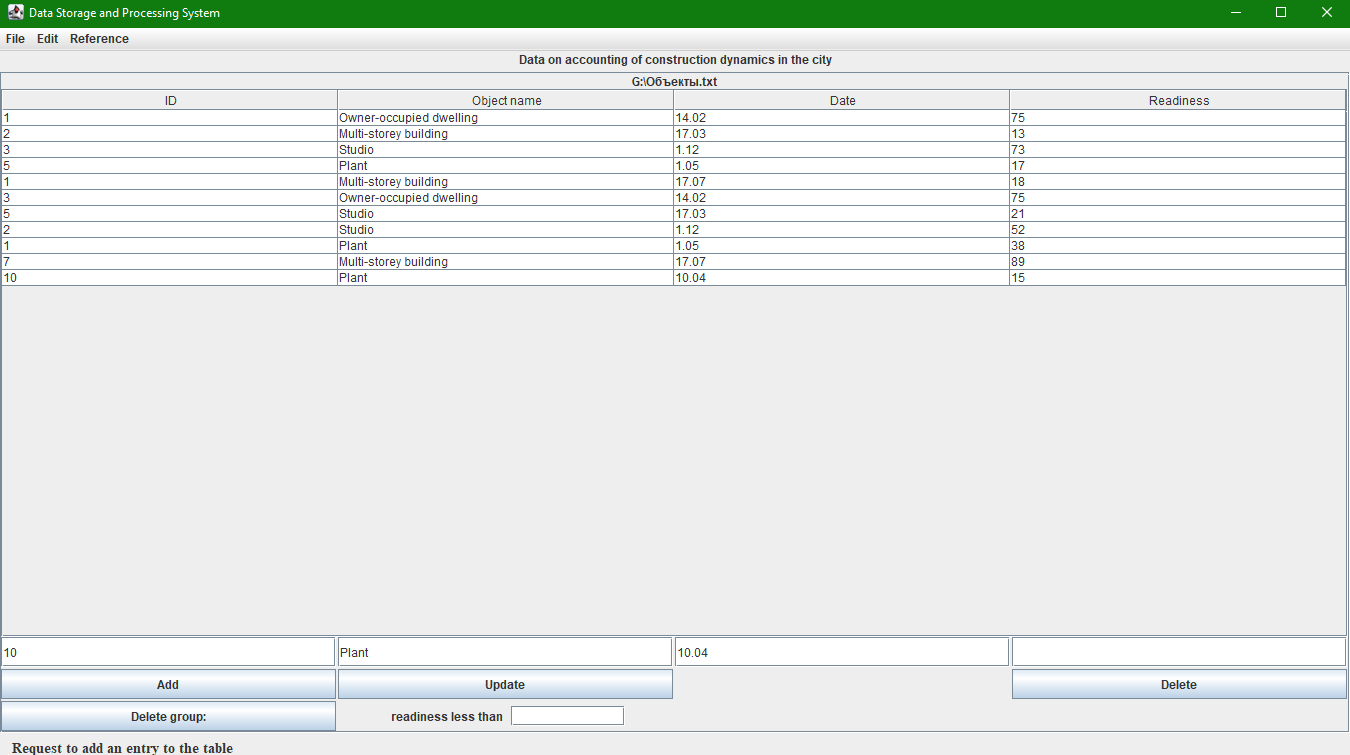


Рисунок 5.13 – добавление записи в таблицу



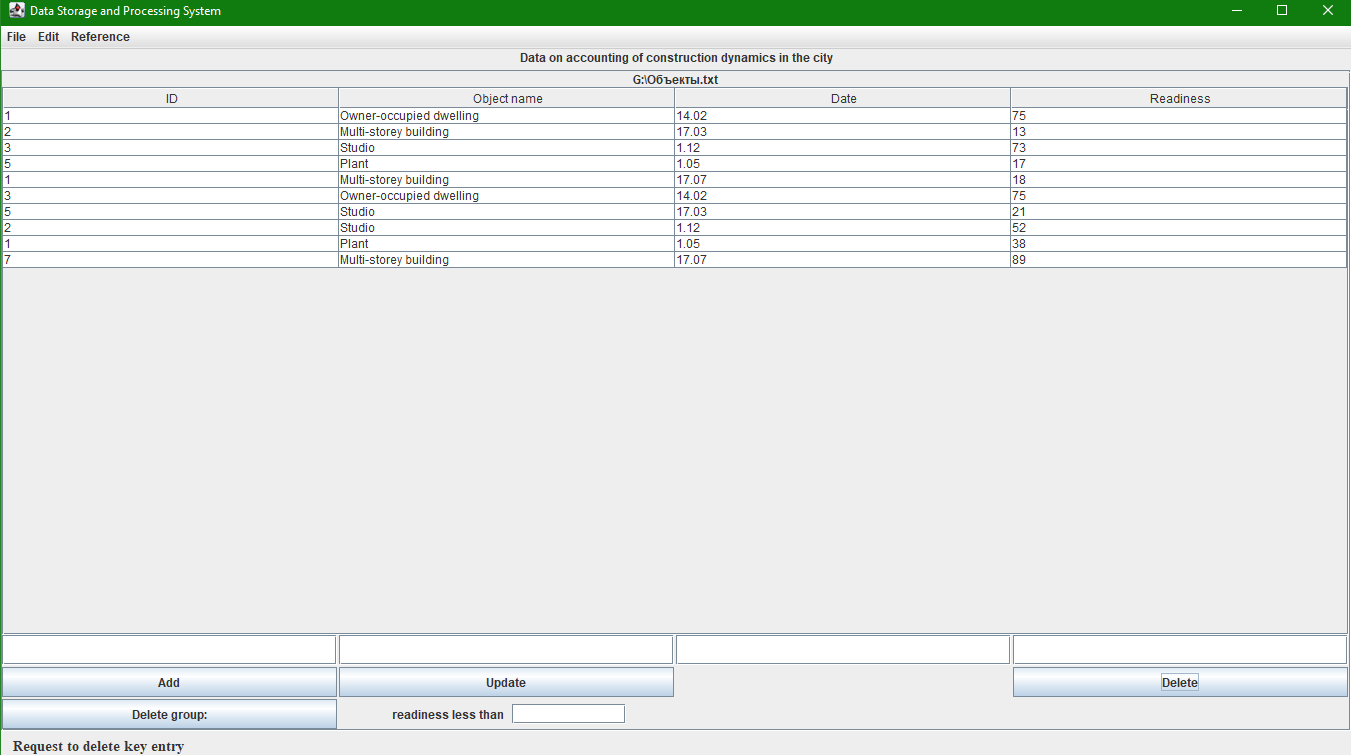
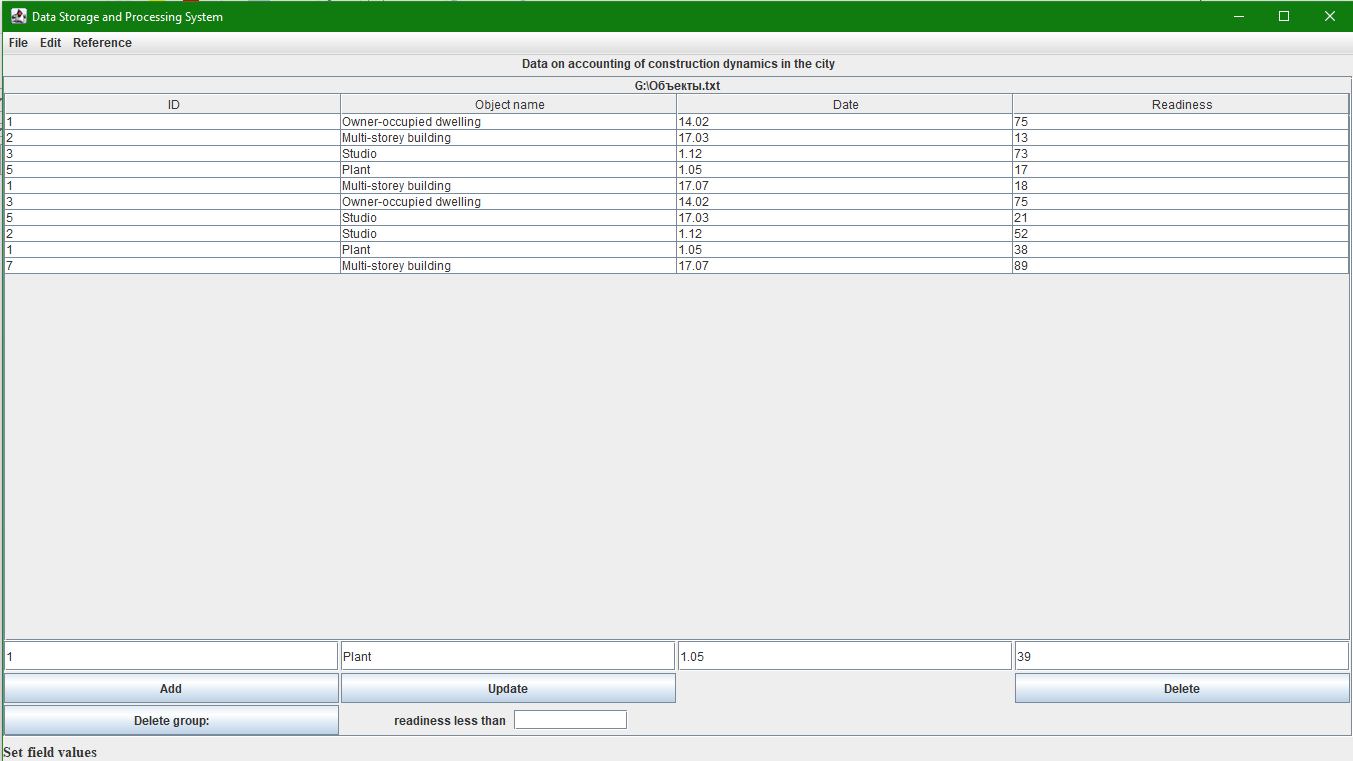


Рисунок 5.14 – удаление записи из таблицы



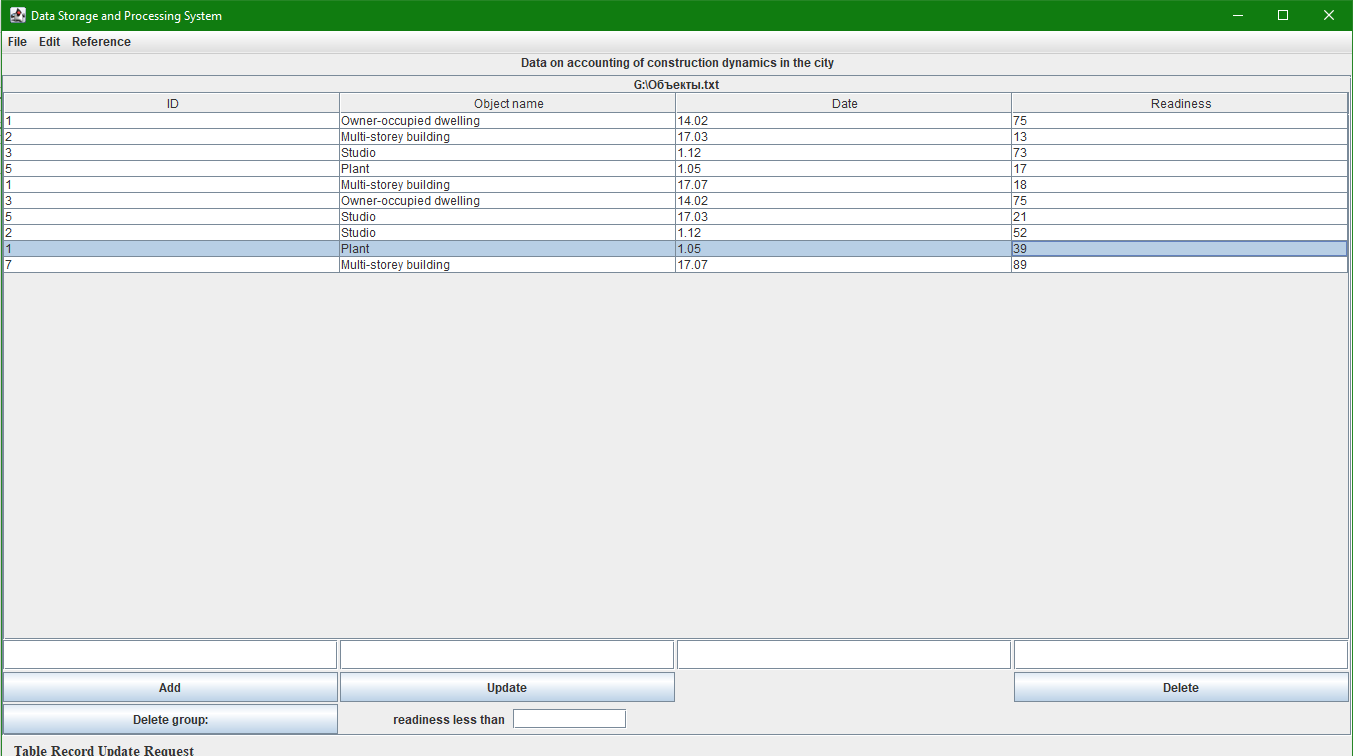
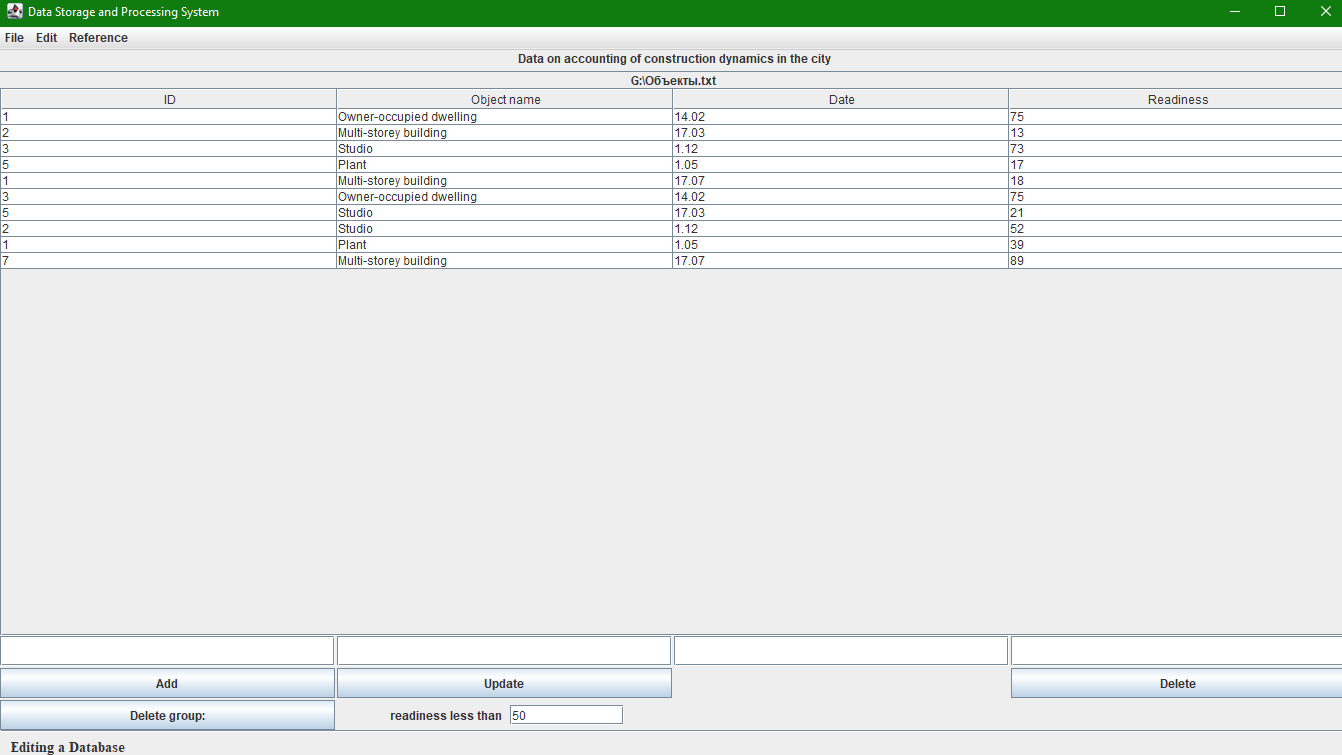


Рисунок 5.15 – редактирование записи



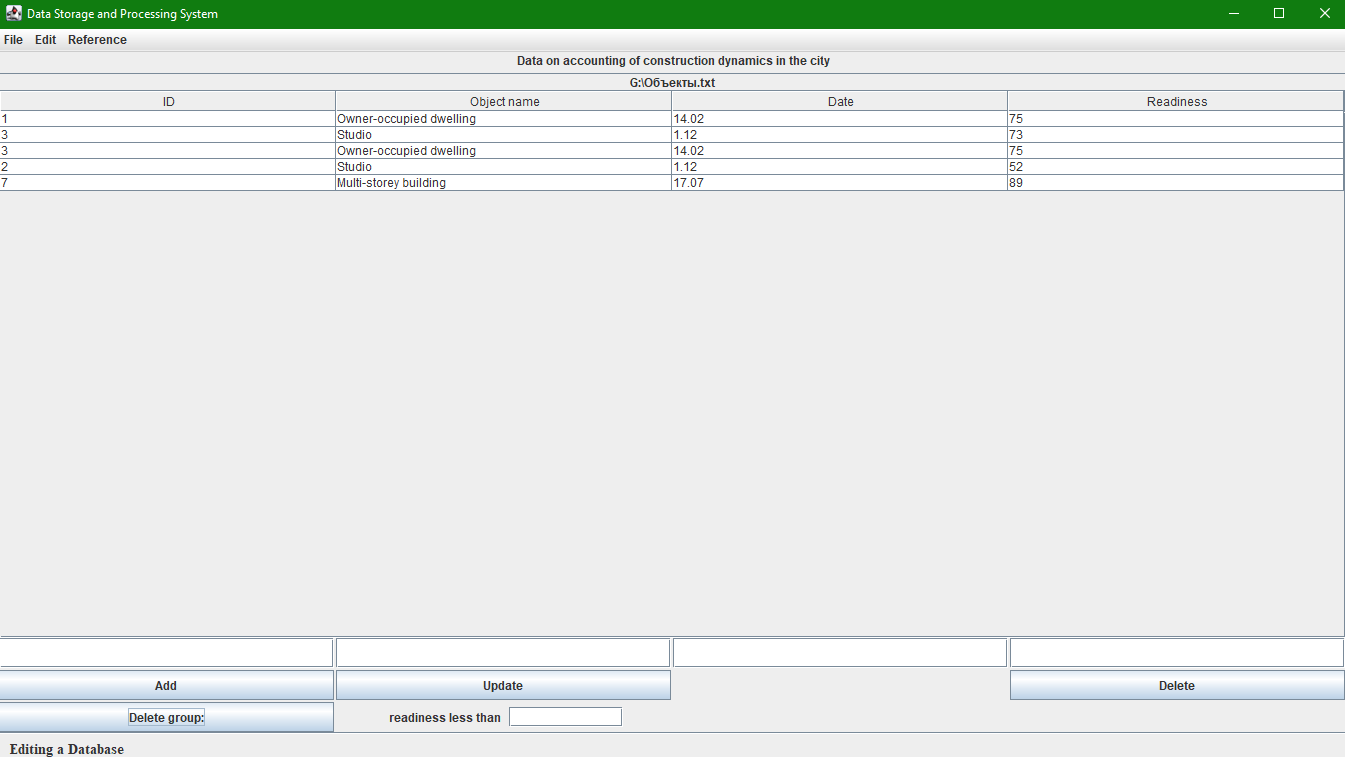


Рисунок 5.16 – удаление группы записей

Процесс сохранения файла показан на рисунках 5.17 и 5.18

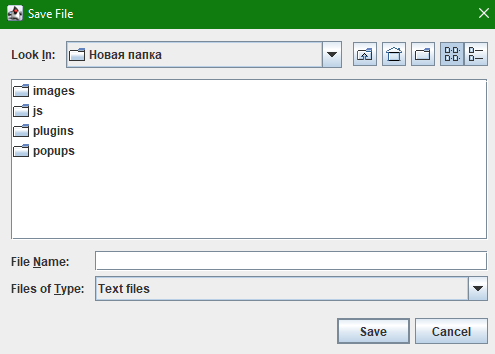


Рисунок 5.17 – Вид главного окна после выбора команды File→Save as

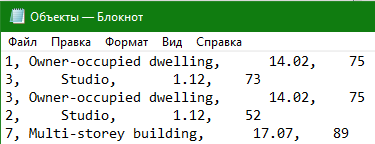


Рисунок 5.18– Сохраненный файл Объекты.txt

Окна программы, открытые при нажатии кнопок во вкладке «Reference» показаны на рисунках 5.19 и 5.20

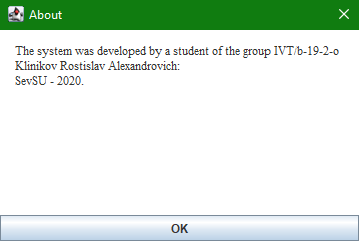


Рисунок 5.19– О программе

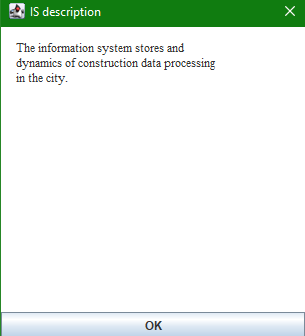


Рисунок 5.20– Описание информационной системы

# **6. ОПИСАНИЕ ПРИМЕНЕНИЯ ПРОГРАММЫ**

Для использования разработанной программы, необходимо иметь операционную систему Windows 7/8/10. Программа использует очень маленький объем системных ресурсов для своего функционирования, поэтому их объем не столь важен.

Для запуска программы необходимо открыть проект в программе IntelliJ IDEA, кликнуть правой кнопкой мыши по классу Main, и выбрать пункт «Run ‘Main.main()’».

После запуска программы откроется окно с пустой базой данных. С помощью панели редактирования можно добавлять, изменять, удалять и удалять группы записей при помощи соответствующих кнопок, отсортировать таблицу, выдать итоговые расчеты и установить фильтрацию данных.

При помощи меню «File» пользователь может открыть существующую базу данных, сохранить отредактированную базу данных в этом же файле или самостоятельно выбрать место сохранения файла, а также закрыть программу при помощи соответствующих подпунктов меню «File».

В меню «Reference» пользователь может получить информацию о разработанной программе.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы была разработана программная система хранения и обработки данных. Для правильного функционирования программы был разработан ряд тестов, искусственно моделировались исключительные ситуации, которые могут возникать в процессе эксплуатации программы пользователем.

Так же были разработаны: схема программы (чертеж) и схема алгоритма расчета среднего балла по каждому предмету (чертеж).

В процессе программирования были изучены тонкости при работе с графическим интерфейсом пользователя и углублены знания в области применения инструмента коллекций JavaCollectionsFramework.

На последних этапах работы была разработана пояснительная записка и презентация для защиты курсового проекта.

В приложение А выведен листинг разработанной программы.

# **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Ноутон П. Java 2 / П.Ноутон, Г.Шилдт. – СПб.: БХВ- Петербург, 2007. –1072. с.

2. Эккель Б. Философия Java / Б. Эккель. – СПб.: Питер, 2015. – 1168 с.

3. Гагарина Л.Г. Технология разработки программного обеспечения: учеб. пособие/ Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. – М.: ИД "ФОРУМ": ИНФРА-М, 2008. – 400 с.

4. Хомоненко А.Д. Базы данных: учебник для высших учебных заведений/ А.Д.Хомоненко, В.М.Цыганков, М.Г.Мальцев. – СПб.:КОРОНА принт, 2009. – 672с.

6. Общие требования к текстовым документам. ГОСТ 2.105—95. ЕСКД. М.: «Стандартинформ», 2005. — 28 с.

7. ГОСТ 2.109-73 Единая система конструкторской документации (ЕСКД). Основные требования к чертежам (с Изменениями N 1-11). М.: Стандартинформ, 2011

8. ЕСПД. ГОСТ 18.701−90 (ИСО 5807-85). Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. – Введ. 1992-01-01. – М: Государственный стандарт Союза ССР.

9. JavaFX 2.0: разработка RIA-приложений. / Тимур Машнин — СПб.: БХВ-Петербург, 2012.-320 с.: ил. — (Профессиональное программирование)

# **ПРИЛОЖЕНИЕ А**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  | СевГУ.2020.ИВТб-19-2-о.34.01 | | | | | | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | Подсистема запуска ГИП | Лит | | | Масса | Масштаб | | Изм | Лист | № Докум. | Подп. | Дата |  |  |  |  |  | | Разраб. | | Клиников Р.А. |  |  | | Руковод. | | Волкова Т.В. |  |  | | Реценз. | |  |  |  | лист 1 | | | Листов 4 | | | Н.Контр. | | Волкова Т.В. |  |  |  | Кафедра ИТиКС. КР по дисциплине  «Программирование. Языки высокого уровня» | | | | | |  | |  |  |  | | Утв. | | Брюховецкий А.А. |  |  |     1  2  3  0   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  | СевГУ.2020.ИВТб-19-2-о.34.02 | | | | | | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | Алгоритм функционирования разрабатываемой системы | Лит | | | Масса | Масштаб | | Изм | Лист | № Докум. | Подп. | Дата |  |  |  |  |  | | Разраб. | | Клиников Р.А. |  |  | | Руковод. | | Волкова Т.В. |  |  | | Реценз. | |  |  |  | лист 2 | | | Листов 4 | | | Н.Контр. | | Волкова Т.В. |  |  |  | Кафедра ИТиКС. КР по дисциплине  «Программирование. Языки высокого уровня» | | | | | |  | |  |  |  | | Утв. | | Брюховецкий А.А. |  |  |      |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  | СевГУ.2020.ИВТб-19-2-о.34.03 | | | | | | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | Алгоритм выполнения пункта меню Открыть | Лит | | | Масса | Масштаб | | Изм | Лист | № Докум. | Подп. | Дата |  |  |  |  |  | | Разраб. | | Клиников Р.А. |  |  | | Руковод. | | Волкова Т.В. |  |  | | Реценз. | |  |  |  | лист 3 | | | Листов 4 | | | Н.Контр. | | Волкова Т.В. |  |  |  | Кафедра ИТиКС. КР по дисциплине  «Программирование. Языки высокого уровня» | | | | | |  | |  |  |  | | Утв. | | Брюховецкий А.А. |  |  | |
| Процент готовности  меньше заданного   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  | СевГУ.2020.ИВТб-19-2-о.34.04 | | | | | | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | Алгоритм удаления группы записей | Лит | | | Масса | Масштаб | | Изм | Лист | № Докум. | Подп. | Дата |  |  |  |  |  | | Разраб. | | Клиников Р.А. |  |  | | Руковод. | | Волкова Т.В. |  |  | | Реценз. | |  |  |  | лист 4 | | | Листов 4 | | | Н.Контр. | | Волкова Т.В. |  |  |  | Кафедра ИТиКС. КР по дисциплине  «Программирование. Языки высокого уровня» | | | | | |  | |  |  |  | | Утв. | | Брюховецкий А.А. |  |  | |