

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

|  |  |  |
| --- | --- | --- |
|  | | |
| Зав. кафедрой | | «ПОВТиАС» |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | В.В. Долгов |
| подпись | |  |
| «\_\_\_\_» | \_\_\_\_\_\_\_\_\_\_ 2024 г. | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(бакалаврская работа)**

Тема: «ВЕКТОРНЫЙ РЕДАКТОР КОМПЬЮТЕРНЫХ СХЕМ И ДИАГРАММ»

Направление подготовки 09.03.04 Программная инженерия

Направленность Программное обеспечение вычислительной техники и автоматизированных систем

Обозначение ВКР 09.03.04.540000.000 Группа ВПР44

Обучающийся М.Л. Лапардин

подпись, дата

Руководитель ВКР доцент В.В. Долгов

подпись, дата

Консультанты по разделам:

Безопасность и экологичность доцент А.Р. Темирканов

подпись, дата

Экономическое обоснование доцент В.Г. Лисицин

подпись, дата

Нормоконтроль И.В. Котельникова

подпись, дата

Ростов-на-Дону

2024



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

|  |  |  |
| --- | --- | --- |
|  | | |
| Зав. кафедрой | | «ПОВТиАС» |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | В.В. Долгов |
| подпись | |  |
| «\_\_\_\_» | \_\_\_\_\_\_\_\_\_\_ 2024 г. | |
|  |  | |

**ЗАДАНИЕ**

на выполнение выпускной квалификационной работы

Тема: «ВЕКТОРНЫЙ РЕДАКТОР КОМПЬЮТЕРНЫХ СХЕМ И ДИАГРАММ»

Обучающийся Лапардин Максим Леонидович

Обозначение ВКР 09.03.04.540000.000 Группа ВПР44

Тема утверждена приказом по ДГТУ от «19» апреля 2024 г. № 1929-ЛС-О

Срок представления ВКР к защите «23» июня 2024 г.

Исходные данные для выпускной квалификационной работы:

Задание на выпускную квалификационную работу; Официальная документация по разработке приложений с использованием библиотеки JointJS https://resources.jointjs.com/tutorial.

Содержание выпускной квалификационной работы

Введение:

Описание сферы использования программного средства, обоснование актуальности темы.

Наименование и содержание разделов:

1. Аналитический обзор предметной области и постановка задачи. В данном разделе описывается предметная область, рассматриваются существующие аналоги разрабатываемого программного средства. Выполняется постановка цели и задач. Формируется список функциональных требований к системе.

2. Алгоритмическое конструирование векторного редактора компьютерных схем и диаграмм. Данный раздел описывает основные алгоритмы, необходимые для реализации функций программного средства. Подробно описана работа всех алгоритмов и приведены соответствующие им блок схемы.

3. Программное конструирование. Раздел описывает причины выбора языка программирования JavaScript, среды разработки Visual Studio Code и инструментов для работы с базой данных, основные подпрограммы.

4. В данном разделе проводится тестирование различных функциональных возможностей программного средства. Тестирование охватывает широкий спектр сценариев использования, включая создание новых схем и диаграмм, редактирование существующих, масштабирование, перемещение и удаление элементов, а также соединение элементов линиями и т.д.

5. Экономическое обоснование работы. Описаны экономические аспекты реализации программного средства. Оценена потенциальная социально-экономическая результативность внедрения разработки. Построен сетевой график работ. Рассчитаны себестоимость разработки и срок окупаемости.

6. Безопасность и экологичность работы. Проанализированы опасные и вредные факторы, оказывающие влияние на разработчика программного средства. Произведен расчет системы искусственного освещения рабочего места.

Заключение:

Разработана система «Векторный редактор компьютерных схем и диаграмм». Приводится общее описание проделанной работы. Делается вывод об успешности разработки.

|  |  |  |
| --- | --- | --- |
| Руководитель ВКР | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | зав. кафедры В.В. Долгов |
| Задание принял к исполнению | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | М.Л. Лапардин |

Аннотация

Отчет включает: страниц - 75; рисунков - 21; таблиц - 14; источников - 8; приложений - 3.

Цель работы – разработка удобного векторного редактора компьютерных схем и диаграмм для повышения эффективности обучения.

Данная работа посвящена созданию векторного редактора компьютерных схем и диаграмм для операционных систем Windows, macOS и linux, который может быть внедрен в образовательный процесс в учебных заведениях и офисах. В работе описывается разработка основных алгоритмов для реализации требуемого функционала, выбор среды разработки и проектирование программного средства. Также описывается набор программных модулей, составляющих редактор компьютерных схем и диаграмм.

Abstract

The report includes: pages-75; figures-21; tables-14; sources-8; attachment-3.

The purpose of the work is to create a vector graphics editor for computer diagrams and charts to enhance the efficiency of learning.

This work is devoted to the development of a vector graphics editor for computer diagrams and charts for the Windows, macOS, linux operating systems, which can later be implemented in the educational process in educational institutions and offices. The development of basic algorithms for implementing the required functionality is described. The development environment is selected and the software tool is designed. A set of software modules that make up the computer diagram and chart editor is described.

**Содержание**

[Введение …………………………………………………………………………..7](#_Toc74595664)

[1 Аналитический обзор предметной области 10](#_Toc74595665)

[1.1 Понятие векторного редактора схем и диаграмм 10](#_Toc74595666)

[1.2 История появления 11](#_Toc74595667)

[1.3 Основные функциональные особенности 12](#_Toc74595668)

[1.4 Преимущества и недостатки 12](#_Toc74595668)

[1.5 Обзор существующих аналогов 12](#_Toc74595668)

1.5.1 Обзор программного средства «Diagrams.net» 15

1.5.2 Обзор программного средства «LucidChart» 17

1.5.3 Обзор программного средства «Microsoft Visio» 18

1.5.4 Обзор программного средства «SmartDraw» **Ошибка! Закладка не определена.**

1.5.5 Обзор программного средства «yEd Graph Editor» 20

[1.6 Постановка задачи 22](#_Toc74595669)  
[1.7 Выводы по главе 30](#_Toc74595677)

[2 Алгоритмическое конструирование векторного редактора компьютерных схем и диаграмм 24](#_Toc74595670)

[2.1 Алгоритм создания фигур 24](#_Toc74595671)

[2.2 Алгоритм перемещения фигур 25](#_Toc74595672)

[2.3 Алгоритм масштабирования фигур 26](#_Toc74595673)

2.4 Алгоритм соединения фигур 27

[2.5 Алгоритм выделения 28](#_Toc74595677)

[2.6 Алгоритм копирования и вставки 29](#_Toc74595677)

2.7 Алгоритм отмены действия 30

2.8 Алгоритм повтора действия 31

2.9 Алгоритм сохранения диаграммы 32

[2.10 Алгоритм открытия диаграммы 33](#_Toc74595679)

[2.11 Выводы по главе 34](#_Toc74595680)

3 Программное конструирование 35

3.1 Обоснование выбора средств разработки 35

3.2 Архитектура приложения 36

3.3 Описание основных подпрограмм 37

3.4 Описание основных функций веб-приложения 38

3.5 Выводы по главе 38

4 Тестирование 39

4.1 Описание процесса функционального тестирования 39

4.2 Выводы по главе 46

[5 Экономическое обоснование работы 47](#_Toc74595712)

[5.1 Основные аспекты реализации работы **Ошибка! Закладка не определена.**](#_Toc74595713)

[5.2 Характеристика ПП с позиции маркетинга **Ошибка! Закладка не определена.**](#_Toc74595714)

[5.3 Стратегический маркетинговый анализ целесообразности применения ПП **Ошибка! Закладка не определена.**](#_Toc74595714)

[5.4 Определение продолжительности работ по разработке ПП **Ошибка! Закладка не определена.**](#_Toc74595716)

[5.5 Определение затрат на разработку и внедрение ПП **Ошибка! Закладка не определена.**](#_Toc74595717)

[5.6 Выводы по главе **Ошибка! Закладка не определена.**](#_Toc74595718)

[6 Безопасность и экологичность при разработке программного средства 61](#_Toc74595721)

[6.1 Площадь и объем рабочих помещений 61](#_Toc74595722)

[6.2 Освещение **Ошибка! Закладка не определена.**](#_Toc74595723)

[6.3 Планировка и оснащение рабочего места **Ошибка! Закладка не определена.**](#_Toc74595725)

[6.4 Отдых глаз **Ошибка! Закладка не определена.**](#_Toc74595726)

[6.5 Вывод по главе **Ошибка! Закладка не определена.**](#_Toc74595727)

[Заключение 69](#_Toc74595728)

[Перечень используемых информационных ресурсов 70](#_Toc74595729)

[Приложение](#_Toc74595730) А. [Техническое задание 73](#_Toc74595731)

[Приложение Б.](#_Toc74595732) [Листинг программы 79](#_Toc74595733)

**Введение**

Схемы и диаграммы являются важными инструментами в современном информационном обществе. Они представляют собой графические модели, которые помогают визуализировать и структурировать информацию для более эффективного понимания и анализа. Отличительной чертой схем и диаграмм является их способность представлять сложные концепции и взаимосвязи в удобной и доступной форме.

Основное различие между схемами и диаграммами проявляется в их функциональности: схемы обычно направлены на описание структуры и элементов, в то время как диаграммы сконцентрированы на обозначении взаимосвязей и отношений между этими элементами. Тем не менее, границы между этими понятиями могут быть размытыми, и выбор того или иного термина может зависеть от контекста или предпочтений автора документации. Ключевым моментом является подбор инструментов для создания схем и диаграмм в зависимости от типа документа, его аудитории и уровня сложности представляемой информации.

Одним из ключевых преимуществ использования схем и диаграмм является их способность упрощать сложные процессы и структуры. Путем графического представления информации можно легче выделить основные компоненты и взаимосвязи между ними, что помогает улучшить понимание и принятие решений.

Схемы и диаграммы нашли широкое применение в различных областях деятельности, таких как наука, техника, бизнес, образование и другие. В научных исследованиях они используются для визуализации данных и результатов экспериментов. В инженерном проектировании они помогают представить технические решения и конструкции. В бизнесе они используются для анализа процессов, составления организационных схем и разработки стратегий. В образовании они являются эффективным средством обучения и иллюстрации учебного материала.

Инструменты для создания и редактирования схем и диаграмм обеспечивают широкий набор функциональных возможностей. Они позволяют пользователю легко создавать и настраивать различные типы графических элементов, изменять их размеры, цвета и стили, а также добавлять текст и аннотации для пояснения содержания.

Схемы и диаграммы также играют важную роль в коммуникации и совместной работе. Они могут быть использованы для представления и обсуждения идей, планирования проектов и принятия коллективных решений. Благодаря графической наглядности, схемы и диаграммы способствуют более эффективному обмену информацией и сотрудничеству.

В целом, схемы и диаграммы являются мощным инструментом для визуализации, анализа и коммуникации информации в различных сферах деятельности. Их использование способствует улучшению понимания, принятию решений и достижению целей в различных областях [1].

Целью выпускной квалификационной работы является облегчение процесса создания и редактирования схем и диаграмм, а также предоставление пользователям широкого набора функциональных возможностей и интуитивно понятного интерфейса для удобства работы с векторной графикой. Редактор позволит пользователям создавать качественные и точные графические схемы и диаграммы с минимальными усилиями, что ускорит и упростит их работу.

В первой главе на основании сравнения существующих аналогов приведен обзор предметной области поставленной задачи, а также сформулированы требования к решению задачи.

Во второй главе были представлены алгоритмы, которые использовались для решения задачи.

В третьей главе описано обоснование выбора языка программирования и представлена структура базы данных, которая была использована для создания программного средства.

В четвертой главе была проведено тестирование программного средства.

1 Аналитический обзор предметной области

В данной главе будет проведен разбор основных особенностей предметной области, истории появления, основные функциональные возможности векторных редакторов схем и диаграмм, а также проанализированы существующие аналогичные программные средства. Сформулированы основные задачи, выполнение которых необходимо для написания работы. Будет сделан вывод о наборе необходимых функциональных характеристик для программного средства.

1.1 Понятие векторного редактора компьютерных схем и диаграмм

Векторный редактор компьютерных схем и диаграмм — это специализированное программное обеспечение, предназначенное для создания, редактирования и анализа векторных графических изображений, таких как схемы, диаграммы, графики и другие визуальные элементы. Основное отличие векторных редакторов от растровых заключается в том, что они работают с математическими объектами (векторами), а не с пикселями, что обеспечивает высокую точность и масштабируемость изображений.

Технические особенности векторных редакторов схем и диаграмм включают в себя ряд ключевых аспектов. Во-первых, это работа с векторной графикой, что обеспечивает высокую четкость и масштабируемость изображений при любом их увеличении или уменьшении размера. Векторные редакторы также обладают многослойной структурой, позволяя разделять элементы диаграммы или схемы на различные уровни для удобства редактирования и управления. Кроме того, они предоставляют разнообразные инструменты редактирования, такие как создание форм, линий, текста и других элементов, что делает процесс создания и редактирования графических материалов более гибким и эффективным. Экспорт и импорт изображений в различные форматы и наличие шаблонов и библиотек элементов дополняют функциональность векторных редакторов, делая их мощным инструментом в работе с графическими данными.

Векторные редакторы нашли широкое применение в научных исследованиях, инженерном проектировании, образовании, бизнес-аналитике, информационной безопасности, веб-дизайне и других областях. Они используются для создания электрических схем, технических чертежей, организационных диаграмм, сетевых графиков, диаграмм потоков данных, архитектурных планов, инфографики и многого другого.

Основное преимущество векторных редакторов схем и диаграмм заключается в их способности упрощать и улучшать процесс визуализации и анализа сложной информации. Они помогают пользователям быстро и эффективно создавать графические материалы, что значительно сокращает время, затрачиваемое на подготовку презентаций, отчетов, инструкций и других документов. Кроме того, они упрощают взаимодействие с данными, делая их более доступными и понятными для аудитории.

Для специалистов в разных областях векторные редакторы схем и диаграмм стали неотъемлемым инструментом, который помогает им эффективно представлять свои идеи и концепции, анализировать данные, принимать решения и делиться информацией в удобной форме [2].

## 1.2 История появления

История векторных редакторов схем и диаграмм уходит корнями в развитие компьютерных технологий и потребностей в визуализации информации. Векторная графика, которая легла в основу этих инструментов, начала активно развиваться во второй половине 20 века вместе с расширением компьютерной техники и возрастанием интереса к графическому дизайну и визуализации данных.

Появление первых векторных редакторов схем и диаграмм связано с поиском способов создания качественных и удобных инструментов для работы с графикой на компьютерах. Одним из первых примеров такого рода программного обеспечения является «Sketchpad», разработанный американским инженером Иваном Сазерлендом в 1963 году в лаборатории MIT. Эта программа позволяла создавать векторные изображения с помощью графического интерфейса с использованием специального пера.

С развитием компьютерных технологий и появлением персональных компьютеров в 1980-е годы появились и первые векторные редакторы, доступные широкому кругу пользователей. Программы такого рода, например, Adobe Illustrator, CorelDRAW и Visio, стали популярными инструментами для создания схем, диаграмм, графиков и других векторных изображений.

С течением времени и с развитием интернет-технологий появились онлайн-сервисы и веб-приложения для работы с векторной графикой. Эти сервисы, такие как Lucidchart, draw.io (переименованный в diagrams.net) и Creately, предоставляют пользователям возможность создавать и редактировать схемы и диаграммы непосредственно в браузере, что делает процесс работы более удобным и доступным.

История появления векторных редакторов схем и диаграмм свидетельствует о постоянном развитии и совершенствовании инструментов для работы с графикой, адаптации их к современным технологическим требованиям и потребностям пользователей [3].

## 1.3 Основные функциональные возможности

Векторные редакторы схем и диаграмм предоставляют широкий спектр функциональных возможностей, которые позволяют пользователям создавать, редактировать и анализировать графические элементы. Вот основные из них:

* создание схем и диаграмм. Векторные редакторы предоставляют инструменты для создания различных типов схем и диаграмм, включая блок-схемы, организационные диаграммы, технические чертежи, сетевые графики и другие;
* редактирование элементов. Пользователи могут легко изменять форму, размер, цвет и другие параметры графических элементов, добавлять текст, стрелки, символы и другие дополнительные элементы;
* группировка и выравнивание. Функции группировки и выравнивания позволяют организовать элементы схемы или диаграммы для создания более четкого и упорядоченного визуального представления;
* использование шаблонов. Векторные редакторы часто предоставляют набор готовых шаблонов для быстрого создания различных типов схем и диаграмм без необходимости создания их с нуля;
* импорт и экспорт данных. Пользователи могут импортировать данные из других программ или форматов файла и экспортировать готовые схемы и диаграммы в различные форматы, такие как PNG, JPEG, PDF, SVG и другие;
* работа с командой. Некоторые векторные редакторы предоставляют функции совместной работы, позволяющие нескольким пользователям работать над одной схемой или диаграммой одновременно и обмениваться комментариями и изменениями в реальном времени;
* автоматическое выравнивание и распределение. Эти функции помогают автоматически выравнивать и распределять элементы схемы или диаграммы для достижения более аккуратного и профессионального внешнего вида;
* создание анимаций и интерактивных элементов. Некоторые векторные редакторы предоставляют возможность добавления анимаций и интерактивных элементов в схемы и диаграммы для создания более динамичного и привлекательного визуального контента;

Эти основные функциональные возможности делают векторные редакторы схем и диаграмм незаменимыми инструментами для визуализации информации в различных областях деятельности [2].

## 1.4 Преимущества и недостатки

Векторные редакторы схем и диаграмм являются мощными инструментами для создания и визуализации графических материалов, однако они также имеют как преимущества, так и недостатки.

Преимущества:

* высокое качество изображений - векторные редакторы используют векторную графику, что позволяет создавать изображения высокого качества, не зависящие от разрешения экрана или устройства просмотра;
* масштабируемость - графические элементы, созданные в векторных редакторах, могут быть легко масштабированы без потери качества, что делает их идеальными для печати на различных носителях;
* гибкость и управляемость - пользователи имеют полный контроль над формой, цветом и другими параметрами графических элементов, что обеспечивает высокую гибкость при создании схем и диаграмм;
* возможности коллаборации - некоторые векторные редакторы предоставляют функции совместной работы, позволяя нескольким пользователям работать над одной схемой или диаграммой одновременно;
* большой выбор инструментов - векторные редакторы предлагают широкий спектр инструментов и функций для создания разнообразных графических материалов, что делает их универсальным инструментом для различных задач.

Недостатки:

* сложность в освоении - некоторые векторные редакторы могут быть сложными для освоения из-за большого количества функций и инструментов, что требует времени и обучения;
* высокая стоимость - некоторые профессиональные векторные редакторы могут иметь высокую стоимость, особенно для коммерческого использования, что может быть недоступно для некоторых пользователей;
* необходимость в компьютере - для работы с векторными редакторами требуется компьютер или другое устройство с подходящим программным обеспечением, что может быть неудобно в некоторых ситуациях;
* ограниченные возможности редактирования фотографий - векторные редакторы не всегда предоставляют широкие возможности для редактирования фотографий и растровых изображений, что может быть недостатком для определенных задач;

Несмотря на некоторые недостатки, векторные редакторы схем и диаграмм остаются незаменимыми инструментами для создания качественных графических материалов в различных областях деятельностих [2].

## 1.5 Обзор существующих аналогов

В сфере векторных редакторов компьютерных схем и диаграмм существует ограниченное количество предлагаемых сервисов. Далее будут описаны следующие аналоги: «Diagrams.net», «LucidChart», «Microsoft Visio», «SmartDraw», «yEd Graph Editor». Оба приложения являются бесплатными и свободными для скачивания пользователями.

### 1.5.1 Обзор программного средства «Diagrams.net»

«Diagrams.net», ранее известный как Draw.io, представляет собой мощный онлайн-инструмент для создания различных видов диаграмм и схем простым и интуитивно понятным способом. Благодаря своей бесплатной и доступной онлайн-платформе, «Diagrams.net» стал популярным выбором среди пользователей, которым необходимо создавать профессиональные и качественные графические представления. Платформа предлагает широкий выбор инструментов, символов и шаблонов, а также поддерживает совместную работу в реальном времени, что делает ее идеальным выбором для командной работы над проектами. Одним из главных преимуществ "diagrams.net" является его бесплатность и возможность использования без установки дополнительного программного обеспечения. Тем не менее, некоторые расширенные функции могут быть доступны только в платной версии, а также могут возникнуть ограничения в поддержке форматов файлов [4].

К основным достоинствам «Diagrams.net» можно отнести следующее:

* полный функционал доступен бесплатно;
* разнообразие диаграмм;
* интеграции с G Suite, Google Drive, Dropbox, OneDrive, Github, GitLab;
* широкий выбор инструментов и шаблонов;
* Поддержка плагинов для популярной платформы для отслеживания ошибок и координации проектов Jira.

К основным недостаткам «Diagrams.net» можно отнести следующее:

* невозможность одновременной работы над одним проектом нескольким пользователям;
* недостаточная гибкость в настройке элементов диаграмм;
* ограниченный выбор форматов экспорта файлов;
* отсутствие некоторых основных категорий шаблонов;
* нет версий приложений для Windows и Mac;
* медленная работа с большими диаграммами.

Интерфейс приложения представлен на рисунке 1.1.

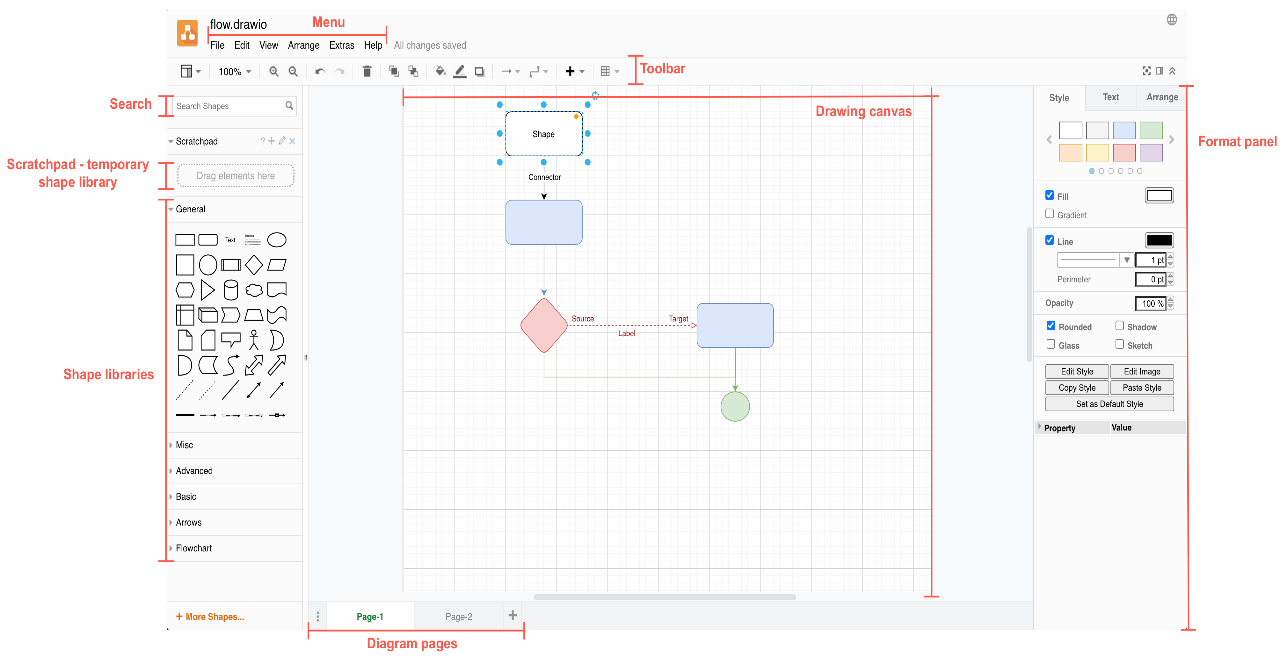


Рисунок 1.1 – Интерфейс приложения «Diagrams.net»

### 1.5.2 Обзор программного средства «LucidChart»

«LucidChart» является онлайн-платформой для создания диаграмм и схем, предоставляющей пользователю широкие возможности для визуализации различных процессов и концепций. Основной функционал включает в себя создание различных типов диаграмм, таких как организационные схемы, блок-схемы, ER-диаграммы, потоковые диаграммы и многие другие. Пользователи могут создавать новые диаграммы с нуля или использовать готовые шаблоны для ускорения процесса. «LucidChart» также предлагает возможность совместной работы над проектами в режиме реального времени, что делает его удобным инструментом для работы в команде [5].

Интерфейс приложения представлен на рисунке 1.2.

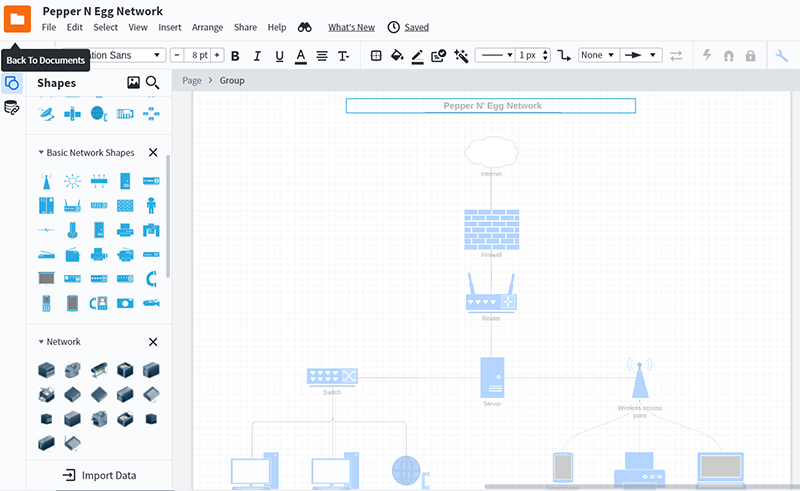


Рисунок 1.2 – Интерфейс приложения «LucidChart»

К основным достоинствам «LucidChart» можно отнести следующее:

* интуитивно понятный интерфейс;
* возможность совместной работы в режиме реального времени;
* широкий выбор инструментов и шаблонов.

К основным недостаткам «LucidChart» можно отнести следующее:

* ограниченные возможности бесплатной версии;
* недостаточная гибкость в настройке элементов диаграмм;
* ограниченный выбор форматов экспорта файлов;
* нет версий приложений для Windows и Mac.

### 1.5.3 Обзор программного средства «Microsoft Visio»

«Microsoft Visio» является программным обеспечением для создания профессиональных диаграмм и схем, разработанным компанией Microsoft. Он предоставляет широкий набор инструментов для создания различных типов диаграмм, включая организационные диаграммы, сетевые диаграммы, планы этажей и т. д. «Microsoft Visio» интегрируется с другими приложениями Microsoft Office, что обеспечивает пользователю единый и удобный интерфейс для работы [6].

К основным достоинствам «Microsoft Visio» можно отнести следующее:

* богатый выбор инструментов для создания различных типов диаграмм;
* интеграция с другими приложениями Microsoft Office;
* возможность работы в офлайн-режиме;
* простота освоения и использования;
* удобный интерфейс и инструменты;
* наличие большого количества шаблонов;
* возможность соединять блок схемы и диаграммы с данными в режиме реального времени.

К основным недостаткам «Microsoft Visio» можно отнести следующее:

* непросто распечатать и опубликовать диаграммы;
* сложность освоения для новичков;
* недоступно для операционных систем macOS и Linux.

Интерфейс приложения представлен на рисунке 1.3.

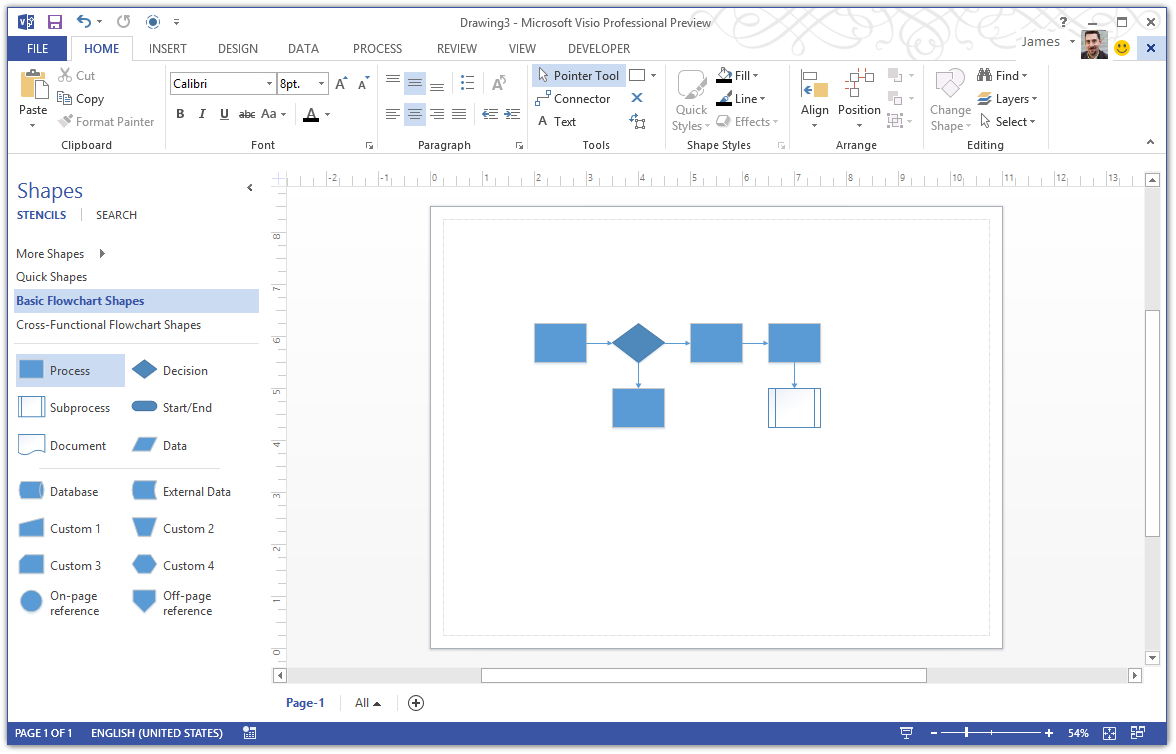


Рисунок 1.3 – Интерфейс приложения «Microsoft Visio»

### 1.5.4 Обзор программного средства «SmartDraw»

«SmartDraw» является программным обеспечением для создания диаграмм и схем, которое предлагает широкий набор инструментов и шаблонов для быстрого создания профессиональных графических материалов. Особенности "SmartDraw" включают в себя автоматическое выравнивание и распределение элементов диаграмм, интеграцию с другими приложениями и сервисами, а также возможность работы как в онлайн-режиме, так и офлайн [7].

К основным достоинствам «SmartDraw» можно отнести следующее:

* интуитивно понятный интерфейс;
* высокая производительность;
* дополнительные функции по типу простого конвертера форматов;
* возможность мгновенных модификаций шаблонов;
* интеграция с другими приложениями и сервисами.

К основным недостаткам «SmartDraw» можно отнести следующее:

* проблемы при соединении диаграмм;
* ограниченный функционал бесплатной версии;
* работает только на Windows.

Интерфейс приложения представлен на рисунке 1.4.

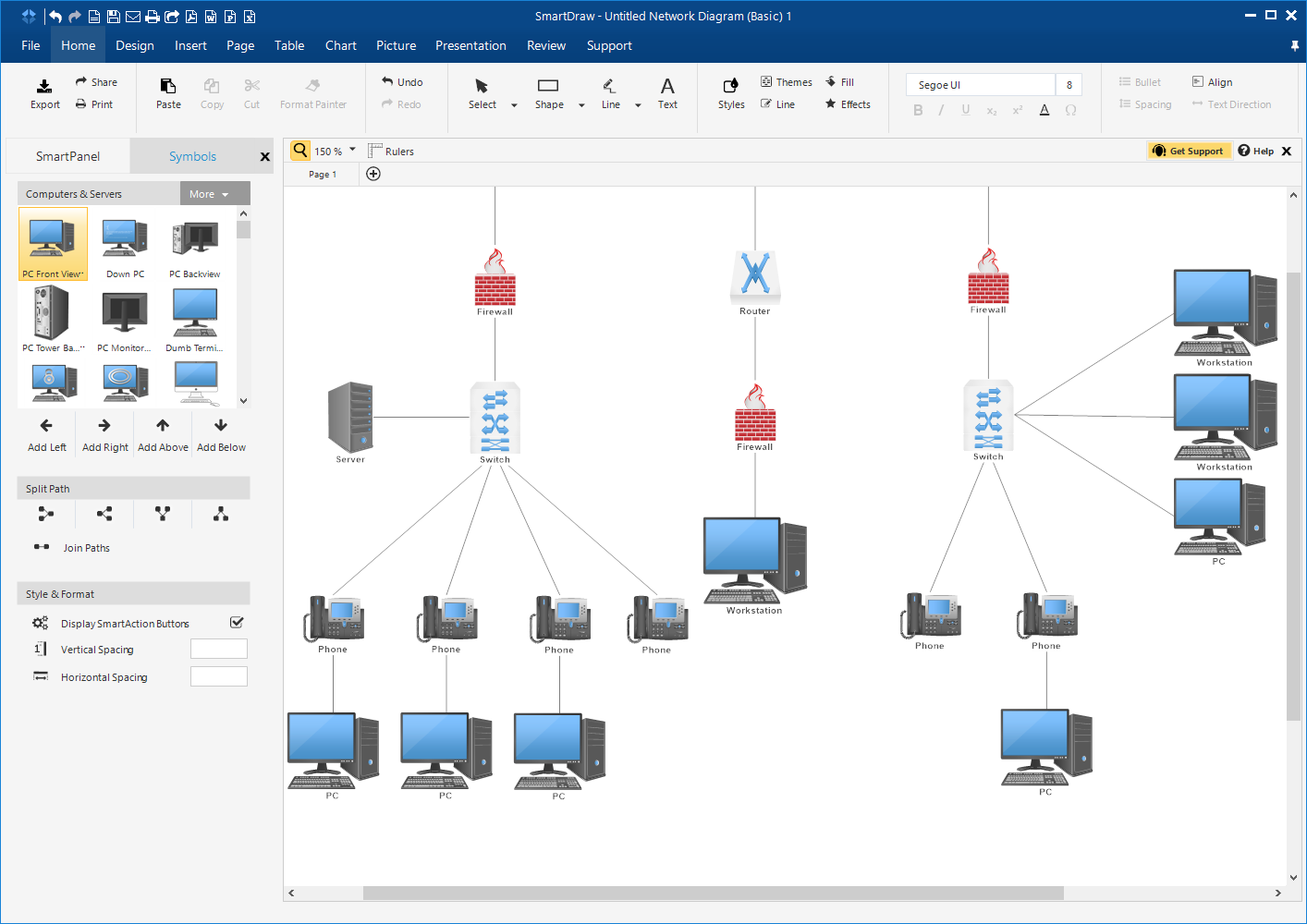


Рисунок 1.4 – Интерфейс приложения «SmartDraw»

### 1.5.5 Обзор программного средства «yEd Graph Editor»

«yEd Graph Editor» является программным обеспечением для создания диаграмм и схем, которое предлагает широкий набор инструментов и шаблонов для быстрого создания профессиональных графических материалов. Особенности "yEd Graph Editor" включают в себя автоматическое выравнивание и распределение элементов диаграмм, интеграцию с другими приложениями и сервисами, а также возможность работы как в онлайн-режиме, так и офлайн [8].

К основным достоинствам «yEd Graph Editor» можно отнести следующее:

* бесплатное использование;
* быстрая и удобная работа со схемами;
* возможность работать с одним и тем же графиком из любой системы;
* наглядная визуализация структуры построения процессов;
* создание схем-каркасов;
* поддержка различных форматов файлов;
* возможность работы в офлайн-режиме.

К основным недостаткам «yEd Graph Editor» можно отнести следующее:

* ограниченные возможности по сравнению с аналогами;
* ограниченная поддержка визуальных стилей и шаблонов;
* отсутствие возможности построения анимированных изображений.

Интерфейс приложения представлен на рисунке 1.5

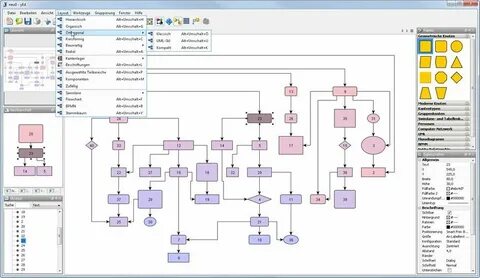


Рисунок 1.5 – Интерфейс приложения «yEd Graph Editor»

## 1.6 Постановка задачи

Проведенный анализ существующих аналогов векторных редакторов схем и диаграмм выявил некоторые ограничения и недостатки. Несмотря на наличие некоторых полезных функций в таких редакторах, они не всегда обеспечивают полное удовлетворение потребностей пользователей и не позволяют полностью реализовать потенциал визуализации и анализа сложных данных и процессов. Таким образом, возникает потребность в разработке собственного векторного редактора схем и диаграмм, который бы предоставил пользователю более широкий и гибкий функционал, а также улучшил пользовательский опыт.

Целью разработки данного векторного редактора схем и диаграмм является создание инструмента, который позволит пользователям эффективно создавать, редактировать и анализировать различные типы графических элементов, таких как схемы, диаграммы, графики и другие векторные изображения. Главной целью является обеспечение пользователей возможностью легко и интуитивно воплощать свои идеи в визуальной форме, с минимальными усилиями и максимальной гибкостью.

Для достижения этой цели необходимо решить следующие задачи:

* разработать интуитивно понятный интерфейс для добавления, редактирования и удаления элементов на схемах и диаграммах;
* разработать современный и стильный дизайн приложения;
* создать набор инструментов, содержащих фигуры и методы работы с ними;
* разработать систему свойств объектов, которая позволит определять характеристики фигур, схем и диаграмм;
* предоставить возможность сохранения и загрузки созданных схем и диаграмм в различных форматах.

## Выводы по главе

В данной главе были определены основные задачи, необходимые для успешной реализации векторного редактора схем и диаграмм. Проведен анализ уже существующих решений в этой области, выявлены их особенности и функциональные возможности. На основе этого анализа был сформулирован список функциональных требований к разрабатываемому программному средству, который будет реализован в рамках выпускной квалификационной работы.

2 Алгоритмическое конструирование векторного редактора компьютерных схем и диграмм

В данной главе приводится описания процесса алгоритмического конструирования разрабатываемого векторного редактора компьютерных схем и диаграмм. Дается описание основных алгоритмов необходимых для реализации заявленных функциональных характеристик разрабатываемого приложения, а также приводятся соответствующие им блок схемы.

2.1 Алгоритм создания фигур

Данный алгоритм обеспечивает пользователю возможность создавать и редактировать разнообразные элементы, которые могут быть использованы для построения диаграмм, схем, и других графических представлений данных.

Схема данного алгоритма представлена на рисунке 2.1.



Рисунок 2.1 — Схема алгоритма создания фигуры

2.2 Алгоритм перемещения фигур

Данный алгоритм используется для перемещения фигур по холсту. Он включает выделение нужной фигуры, захват курсором и перетаскивание по холсту, а также обновление координат фигуры при отпускании курсора.

Схема данного алгоритма представлена на рисунке 2.2.

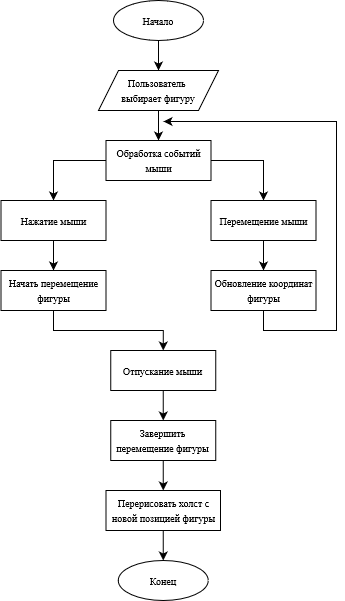


Рисунок 2.2 — Схема алгоритма перемещения фигур

2.3 Алгоритм масштабирования фигур

Данный алгоритм программе представляет собой процесс изменения размера графических объектов на холсте. Пользователь выбирает фигуру и взаимодействует с соответствующими элементами управления, например, масштабирующими ручками или командами. При этом фигура пропорционально изменяет свои размеры, а также могут корректироваться другие параметры, связанные с ее внешним видом. Схема данного алгоритма представлена на рисунке 2.3.

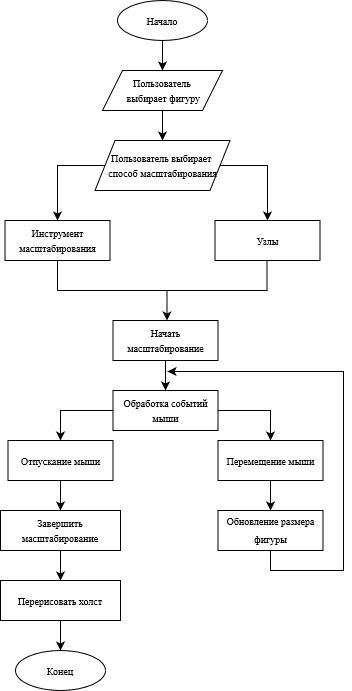


Рисунок 2.3 — Схема алгоритма масштабирования фигур

2.4 Алгоритм соединения фигур

Алгоритм представляет собой процесс установления связей между различными графическими объектами на холсте. Пользователь выбирает начальную и конечную точки соединения на фигурах и инициирует процесс соединения. Программа автоматически создает линии или другие соединительные элементы, визуально связывая выбранные фигуры. Схема данного алгоритма представлена на рисунке 2.4.

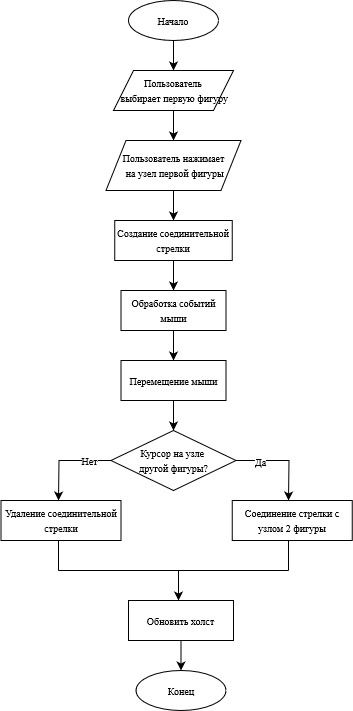


Рисунок 2.4 — Схема алгоритма соединения фигур

2.5 Алгоритм выделения

Алгоритм предназначен для выделения графических объектов на холсте при взаимодействии пользователя с интерфейсом. При наведении курсора мыши на объект, программа отображает контур выделения вокруг него. При клике на объекте или перетаскивании мыши по области объекта, программа активирует его выделение, отображая специальные управляющие элементы, такие как маркеры изменения размера или инструменты для манипулирования объектом. Схема данного алгоритма представлена на рисунке 2.5.



Рисунок 2.5 — Схема алгоритма выделения

2.6 Алгоритм копирования и вставки

Алгоритм предназначен для создания копий выбранных графических объектов и их последующего размещения в других областях холста. Пользователь может выбрать один или несколько объектов и выполнить команду копирования. Затем, переместив курсор мыши в требуемую позицию, пользователь выполняет команду вставки, что приводит к появлению копий выбранных объектов в новом месте на холсте. Схема данного алгоритма представлена на рисунке 2.6.

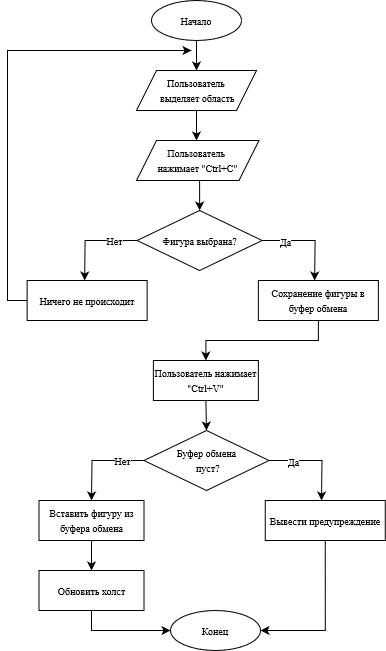


Рисунок 2.6 — Схема алгоритма копирования и вставки

2.7 Алгоритм отмены действия

Данный алгоритм предназначен для отмены последнего выполненного пользователем действия. При активации этой функции программа откатывает текущее состояние к предыдущему, возвращая все изменения, сделанные пользователем, к предыдущему состоянию. Это позволяет пользователям исправить ошибки или вернуться к предыдущему состоянию документа без необходимости открывать и сохранять предыдущие версии файла. Схема данного алгоритма представлена на рисунке 2.7.

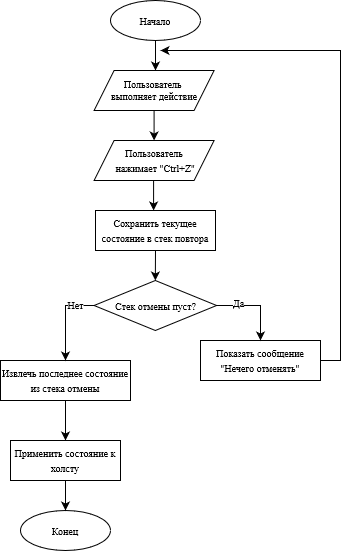


Рисунок 2.7 — Схема алгоритма отмены действия

2.8 Алгоритм повтора действия

Данный алгоритм служит для восстановления отмененных пользователем последних изменений. При активации этой функции программа повторяет последнее отмененное действие, возвращая документ к состоянию, которое было до отмены. Это позволяет пользователям вернуть отмененные изменения или выполнить дополнительные операции после отмены, не теряя результатов своей работы.

Схема данного алгоритма представлена на рисунке 2.8.

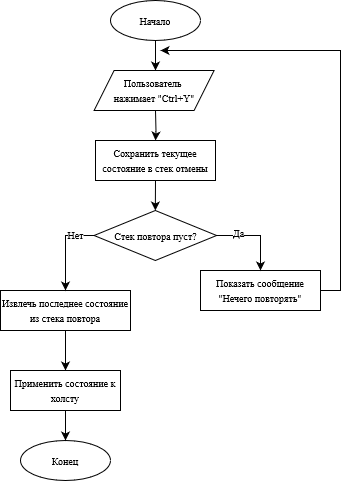


Рисунок 2.8 — Схема алгоритма повтора действия

2.9 Алгоритм сохранения диаграммы

Алгоритм представляет собой процесс преобразования данных текущего состояния диаграммы в формат, который можно сохранить на устройстве пользователя. Сначала программа собирает информацию о расположении и свойствах всех элементов диаграммы, включая их размеры, цвета и текст. Затем эти данные конвертируются в определенный формат файлов (JSON, SVG, JPEG или PNG), который соответствует требованиям пользователя или стандартам программы. После этого сформированный файл сохраняется на устройстве пользователя, готовый для последующего открытия и просмотра. Схема данного алгоритма представлена на рисунке 2.9.

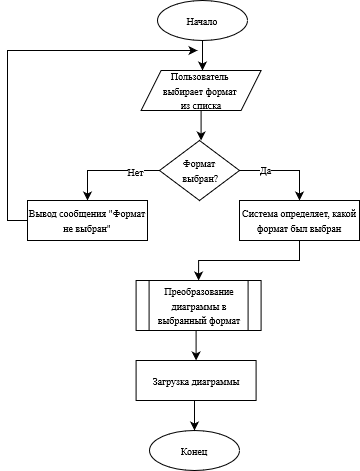


Рисунок 2.9 — Схема алгоритма сохранения диаграммы

2.10 Алгоритм открытия диаграммы

Алгоритм открытия диаграммы в программе начинается с выбора файла пользователем через интерфейс программы. Затем программа считывает выбранный файл с диаграммой с устройства пользователя. После чтения файла программа анализирует его содержимое и извлекает данные о расположении элементов, их свойствах и взаимных связях. Полученная информация используется для воссоздания диаграммы в программе с сохранением всех характеристик элементов. В результате алгоритма пользователь видит открытую диаграмму на экране, которая точно соответствует ранее сохраненной версии, готовую к дальнейшему редактированию или просмотру. Схема данного алгоритма представлена на рисунке 2.10.



Рисунок 2.10 — Схема алгоритма открытия диаграммы

2.11 Выводы по главе

В данной главе были приведены блок схемы основных алгоритмов необходимых для работы разрабатываемого векторного редактора компьютерных схем и диаграмм, а именно следующие алгоритмы:

* алгоритм создания фигур;
* алгоритм перемещения фигур;
* алгоритм масштабирования;
* алгоритм соединения фигур;
* алгоритм выделения;
* алгоритм копирования и вставки;
* алгоритм отмены действия;
* алгоритм повтора действия;
* алгоритм сохранения диаграммы;
* алгоритм открытия диаграммы.

# 3 Программное конструирование векторного редактора компьютерных схем и диаграмм

В данном разделе будут обоснованы выбор языка программирования, используемый для реализации программного средства и основания выбора среды программирования. Будут определены и описаны основные классы разрабатываемого приложения и структура проекта.

## 3.1 Обоснование выбора средств разработки

Для разработки программного средства векторного редактора схем и диаграмм были выбраны следующие инструменты и технологии. В качестве основного инструмента разработки была выбрана среда разработки VS Code, обеспечивающая широкий набор функций и удобную среду для написания кода на JavaScript [9]. VS Code позволяет удобно организовывать проекты, использовать различные расширения для повышения производительности и обладает высокой степенью настраиваемости [10]. Для реализации клиентской части программного средства использовался язык JavaScript, который широко применяется для разработки веб-приложений благодаря своей универсальности и высокой производительности. JavaScript обеспечивает возможность создания интерактивных и динами

ческих пользовательских интерфейсов. В качестве инструмента для управления версиями и совместной работы над проектом использовался Git, интегрированный в среду разработки VS Code [11]. Git обеспечивает эффективное управление изменениями в коде, позволяет отслеживать и фиксировать изменения, а также удобно совместно работать над проектом. Для визуального представления схем и диаграмм в программном средстве используется библиотека JointJS [12]. Она предоставляет мощные инструменты для создания интерактивных векторных элементов с использованием SVG [13]. JointJS обеспечивает высокое качество изображений и возможность масштабирования без потери качества благодаря использованию векторной графики. Библиотека позволяет легко создавать сложные графические элементы, включая фигуры, связи и метки, а также предоставляет инструменты для их редактирования и взаимодействия. JointJS также поддерживает анимации и интерактивные объекты, что делает его идеальным выбором для создания динамических и привлекательных схем и диаграмм в нашем приложении.

## 3.2 Архитектура приложения

Архитектура векторного редактора компьютерных схем и диаграмм состоит из следующих компонентов:

1. Клиентская часть — это пользовательский интерфейс, который обеспечивает взаимодействие пользователя с приложением. Векторный редактор отображает элементы интерфейса, такие как инструменты рисования, панели инструментов и область холста для создания и редактирования схем и диаграмм;

2. Ядро приложения – это компонент, ответственный за обработку действий пользователя, управление элементами схемы и диаграммы, а также взаимодействие с другими частями приложения, такими как серверная часть;

3. Редактор схем – это основной модуль приложения, который обеспечивает возможность создания, редактирования и отображения схем и диаграмм на холсте. Редактор содержит логику для добавления, перемещения, изменения размера и соединения фигур, а также для преобразования схемы в различные форматы.

4. Серверная часть – это компонент, который обрабатывает запросы от клиентской части приложения, предоставляет доступ к данным и обеспечивает взаимодействие с базой данных для хранения информации о схемах и диаграммах.

Архитектура приложения представлена на рисунке 3.1.

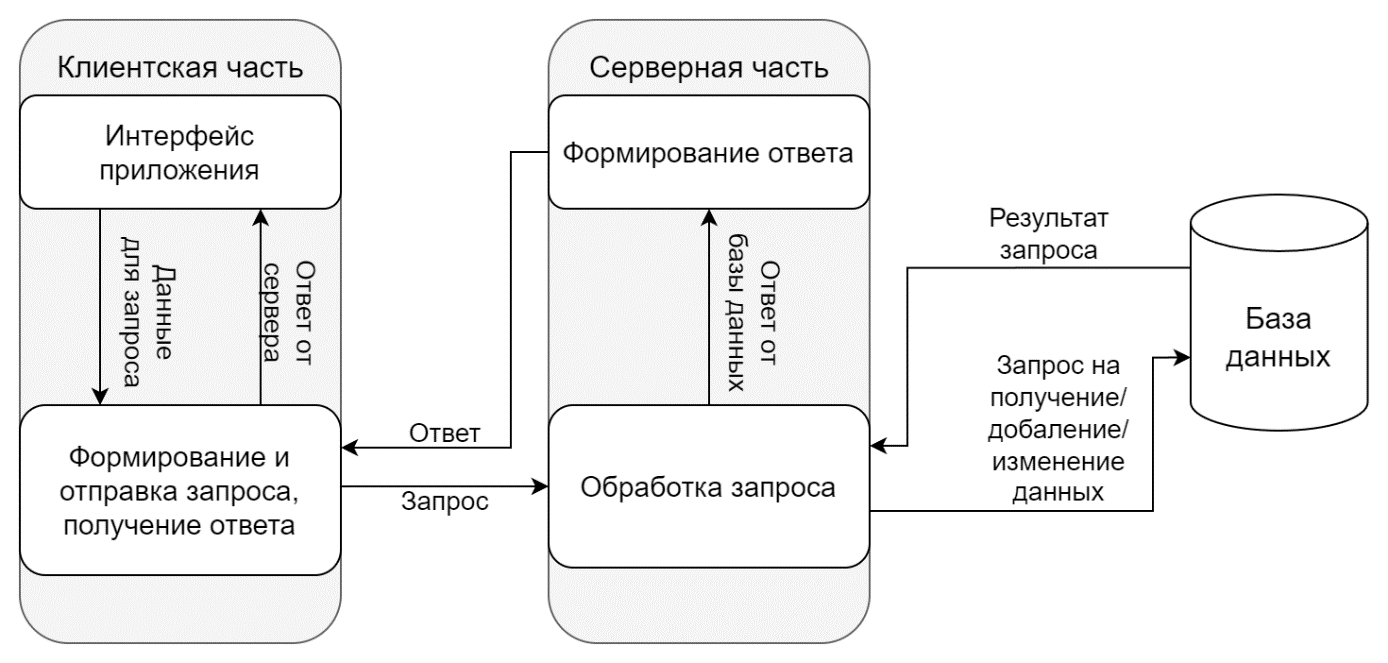


Рисунок 3.1 – Архитектура приложения

## 3.3 Описание основных подпрограмм

Для реализации сформулированных выше алгоритмов необходимо спроектировать и реализовать структуру программного средства. Структура программы представляет из себя набор директорий, которые содержат файлы определённого типа. Рассмотрим наиболее важные для работы программного средства подпрограммы.

В таблице 3.1 представлены подпрограммы из директории html.

Таблица 3.1 – Подпрограммы из директории html

|  |  |
| --- | --- |
| Директория | html |
| Описание директории | Содержит html страницы |
| Подпрограммы | Описание подпрограммы |
| index.html | отвечает за вид главной страницы |

В таблице 3.2 представлены подпрограммы из директории css.

Таблица 3.2 - Подпрограммы из директории css

|  |  |
| --- | --- |
| Директория | css |
| Описание директории | содержит файлы с дизайном приложения |
| Подпрограммы | Описание подпрограммы |
| style.css | отвечает за вид главной страницы |
| menubar.css | отвечает за вид шапки сайта |
| toolbar.css | отвечает за вид панели инструментов |
| properties-panel.css | отвечает за вид панели свойств |
| sidebar.css | отвечает за вид бокового меню |

В таблице 3.3 представлены подпрограммы из директории js.

Таблица 3.3 - Подпрограммы из директории js

|  |  |
| --- | --- |
| Директория | js |
| Описание директории | содержит файлы со всеми алгоритмами |
| Подпрограммы | Описание подпрограммы |
| index.js | отвечает за основные алгоритмы |
| script.js | отвечает за работу бокового меню |

Подробный листинг вышеперечисленных подпрограмм приведен в приложении A.

## 3.4 Основные функции веб-приложения

В таблице 3.4 представлены основные функции веб-приложения, которые находятся в файле «index.js».

Таблица 3.4 – Основные функции веб-приложения

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Входные параметры | Описание | Возвращаемое значение |
| 1 | 2 | 3 | 4 |
| initializeKeyboardShortcuts | - | Устанавливает клавиатурные сокращения для основных операций редактирования диаграммы. Нажатие сочетания клавиш выполняет определенное действие, такое как копирование, вставка, вырезание, отмена или повтор действий, а также выбор всех элементов на холсте. Например, нажатие Ctrl+C копирует выбранные элементы. | - |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| createElement | Строка «shape», которая содержит название элемента | Создает новый элемент на холсте в зависимости от значения строки «shape» | - |
| createToolsForElement | Объект элемента | Cоздает и добавляет инструменты редактирования к элементам и связям на холсте. Эти инструменты включают кнопку информации, кнопку удаления, границу элемента, узлы для связей, а также инструменты для изменения вершин и сегментов связей. Кроме того, функция добавляет обработчики событий для взаимодействия с элементами, такие как отображение инструментов при наведении мыши и редактирование текста по двойному щелчку | - |
| autoSize | Объект элемента | Отслеживает изменения атрибутов элемента и автоматически изменяет его размер в соответствии с содержимым. Это позволяет обеспечивать лучшую визуальную компоновку диаграммы | - |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| exportSVG | - | Экспортирует текущее состояние холста в формат SVG | - |
| exportPNG | - | Экспортирует текущее состояние холста в формат PNG | - |
| exportJPEG | - | Экспортирует текущее состояние холста в формат JPEG | - |
| loadJSONFromFile | - | Загружает данные из JSON файла в граф | - |
| saveJSONToFile | Данные json и имя файла | Сохраняет данные графа в файл JSON | - |
| initializeToolbar | - | Назначает действия для всех кнопок на панели инструментов | - |
| initializePropertiesPanel | - | Назначает действия для всех кнопок на панели свойств | - |
| initializeSidebar | - | Назначает действия для всех кнопок в боковом меню | - |

## 3.5 Выводы по главе

В данной главе был обоснован выбор языка программирования, а также

среда программирования для создаваемого программного средства, описан процесс формирования структуры и архитектуры. Кроме того, были описаны основные подпрограммы программного средства. В таблице было приведено описание каждой из подпрограмм с указанием типа входных параметров и возвращаемого значения.

# 4. Тестирование векторного редактора компьютерных схем и диаграмм

В данном разделе продемонстрированы основные этапы тестирования

векторного редактора компьютерных схем и диаграмм. Приводятся скриншоты, подтверждающие корректность выполнения реализуемых функций.

## 4.1 Описание процесса функционального тестирования

При запуске приложения на экране отображается главная страница. Она содержит в себе шапку, панель инструментов, панель свойств, боковое меню и полотно для фигур. На рисунке 4.1 изображена главная страница.

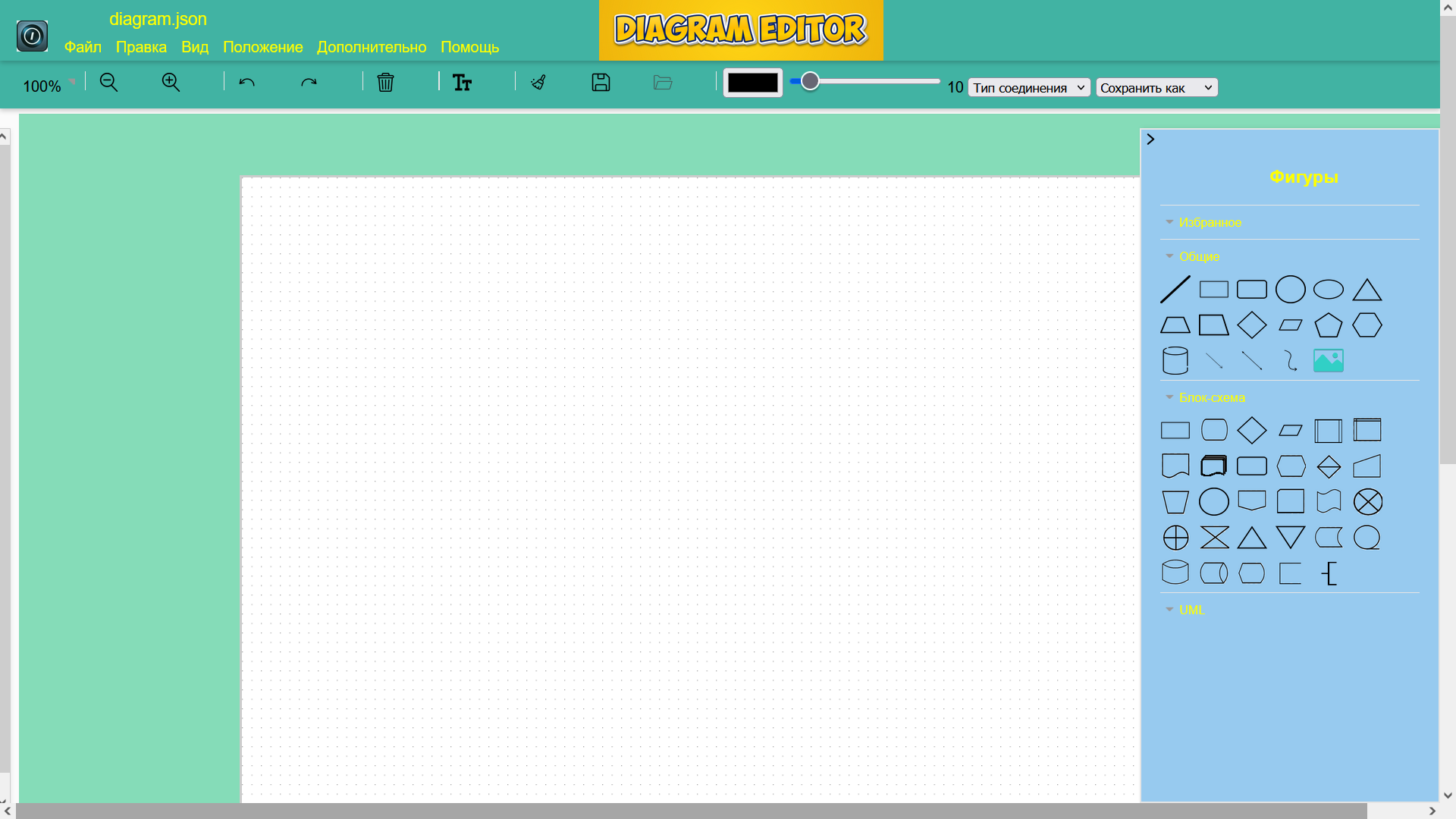


Рисунок 4.1 – Главная страница

Чтобы создать фигуру, нужно нажать на соответствующую кнопку в боковом меню. После этого на полотне появится выбранная фигура, которую можно перемещать, изменять её размер и другие свойства. Процесс создания фигур интуитивно понятен и не требует специальных навыков. На рисунке 4.2 изображено создание нескольких фигур.

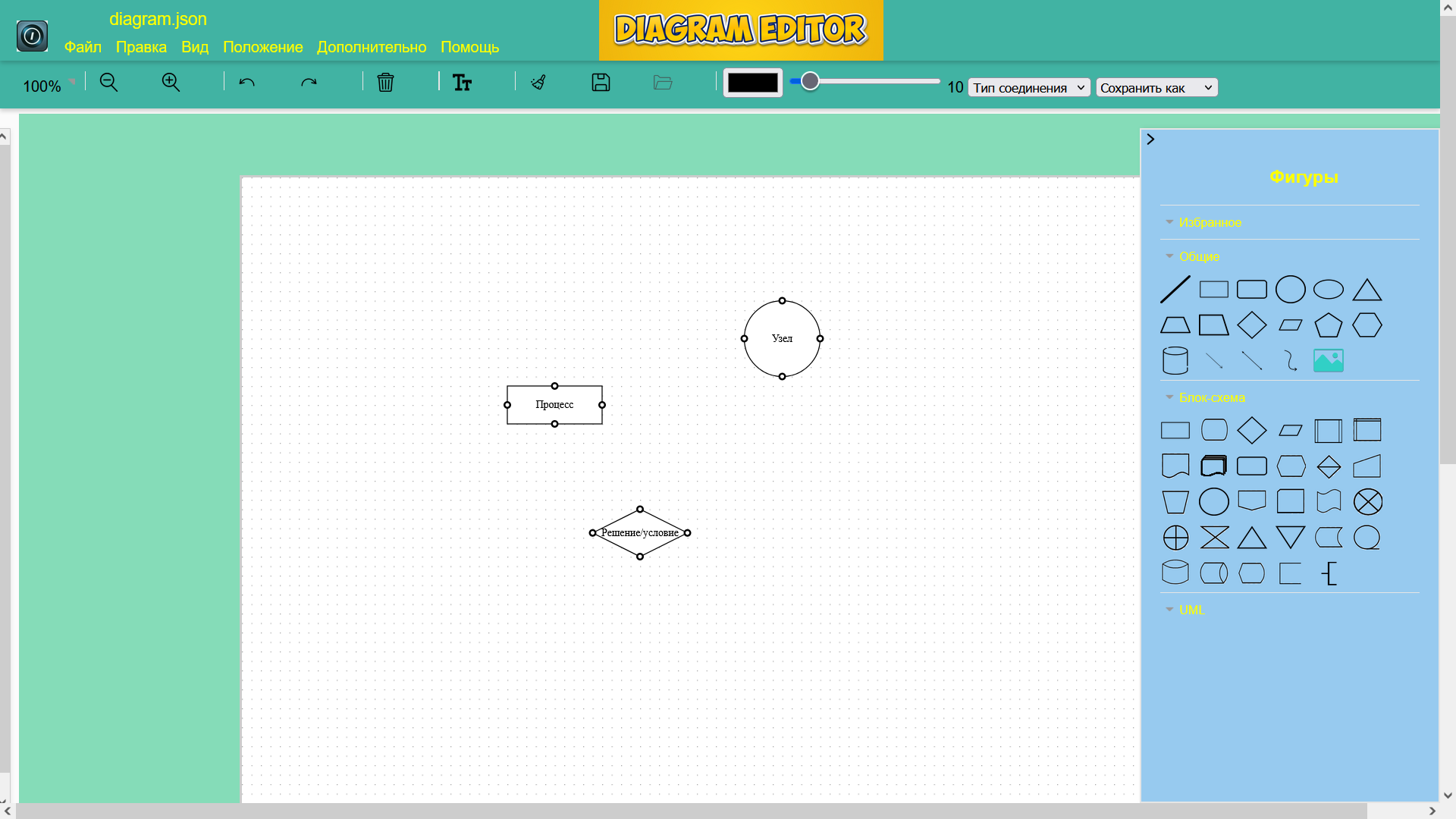


Рисунок 4.2 – Создание фигур

Чтобы переместить фигуру, нужно нажать навести на нее курсор и зажать левую кнопку мыши. При наведении на фигуру также отобразится небольшая панель инструментов и панель свойств. На рисунке 4.3 изображено перемещение фигуры.

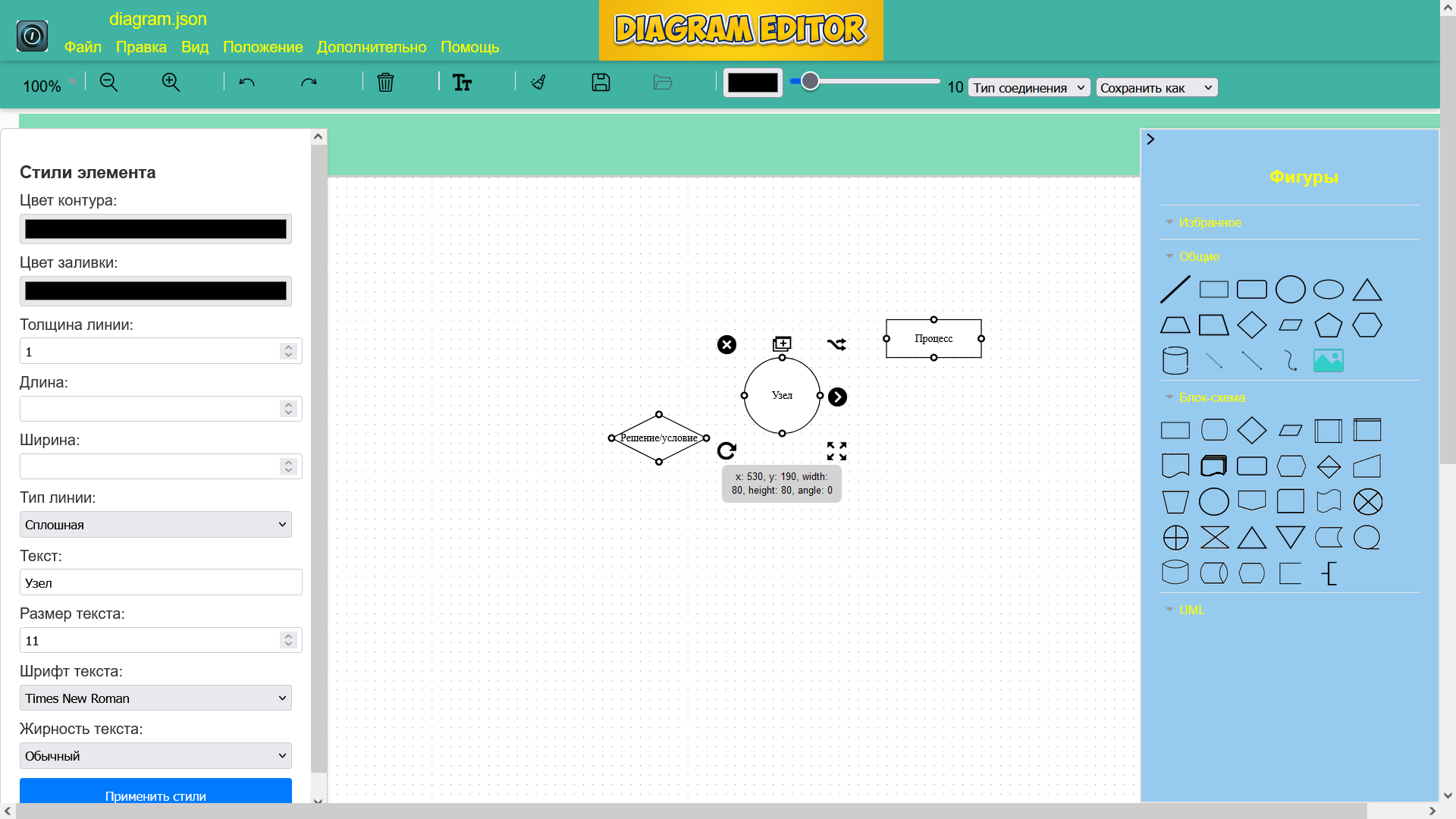


Рисунок 4.3 – Перемещение фигуры

Чтобы изменить размеры фигуры, нужно нажать на фигуру. Отобразится панель инструментов, в правом нижнем углу будет инструмент масштабирования. На рисунке 4.4 изображено масштабирование фигуры.

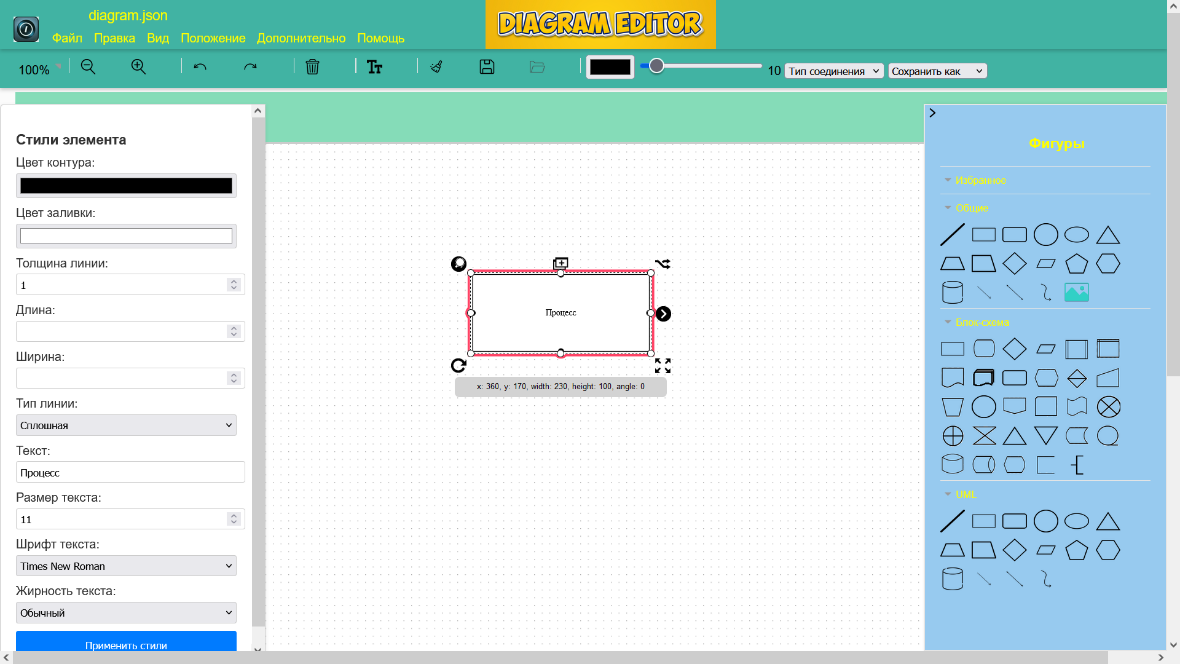


Рисунок 4.4 – Масштабирование фигуры

Чтобы повернуть фигуру, нужно нажать на фигуру и в панели инструментов выбрать инструмент поворота в левом нижнем углу. На рисунке 4.5 изображен поворот фигуры.

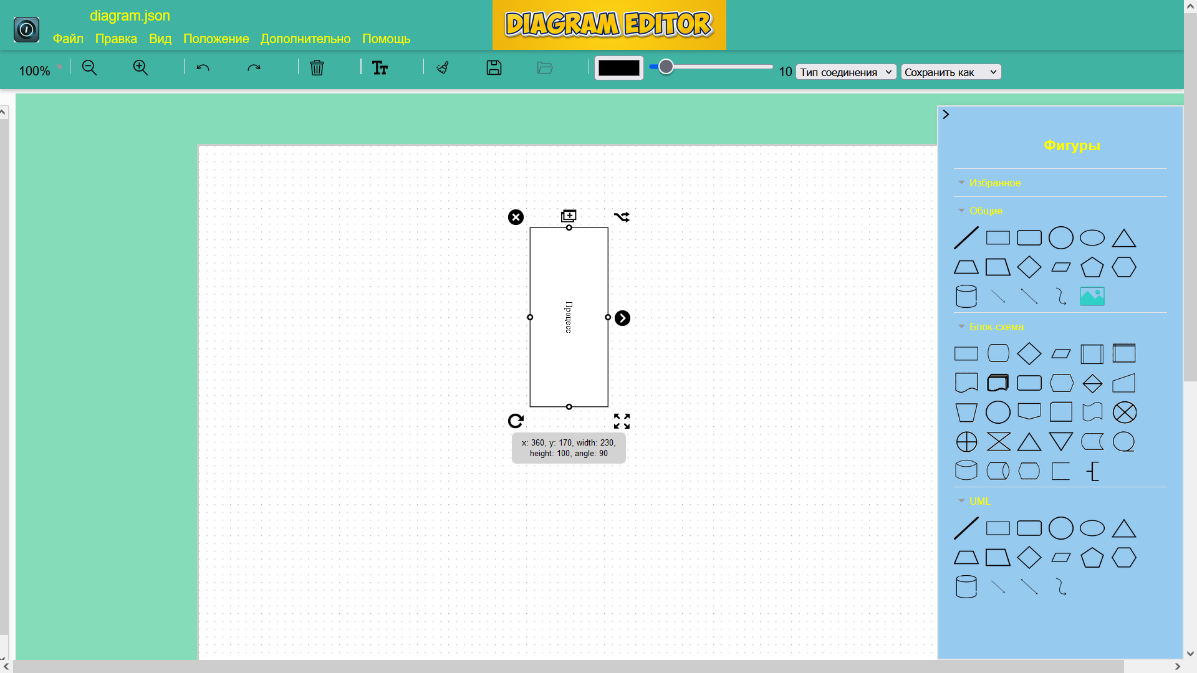


Рисунок 4.5 – Поворот фигуры

Для соединения фигур нужно навести курсор на любой из 4 узлов первой фигуры, нажать левую кнопку мыши и передвинуть курсор к любому из 4 узлов второй фигуры. На рисунке 4.6 изображено соединение фигур.

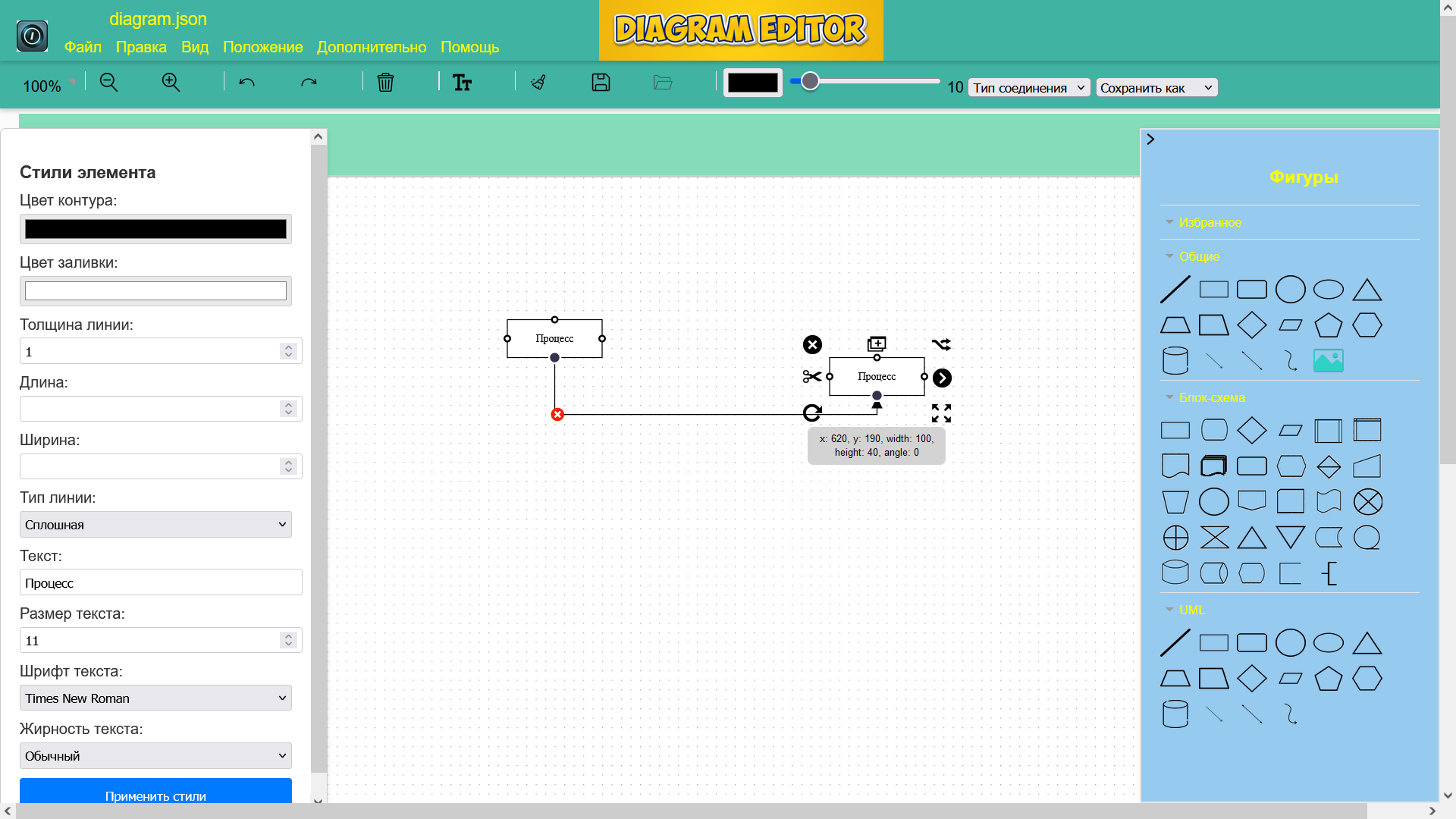


Рисунок 4.6 – Соединение фигур

Чтобы выделить область, нужно нажать левую кнопку мыши в любом месте и растянуть прямоугольник выделения на нужную длину и высоту. Если внутри области есть фигуры, они добавятся в область. Выделенную область с фигурами можно передвинуть, удалить, масштабировать и повернуть. На рисунке 4.7 изображено выделение фигур.

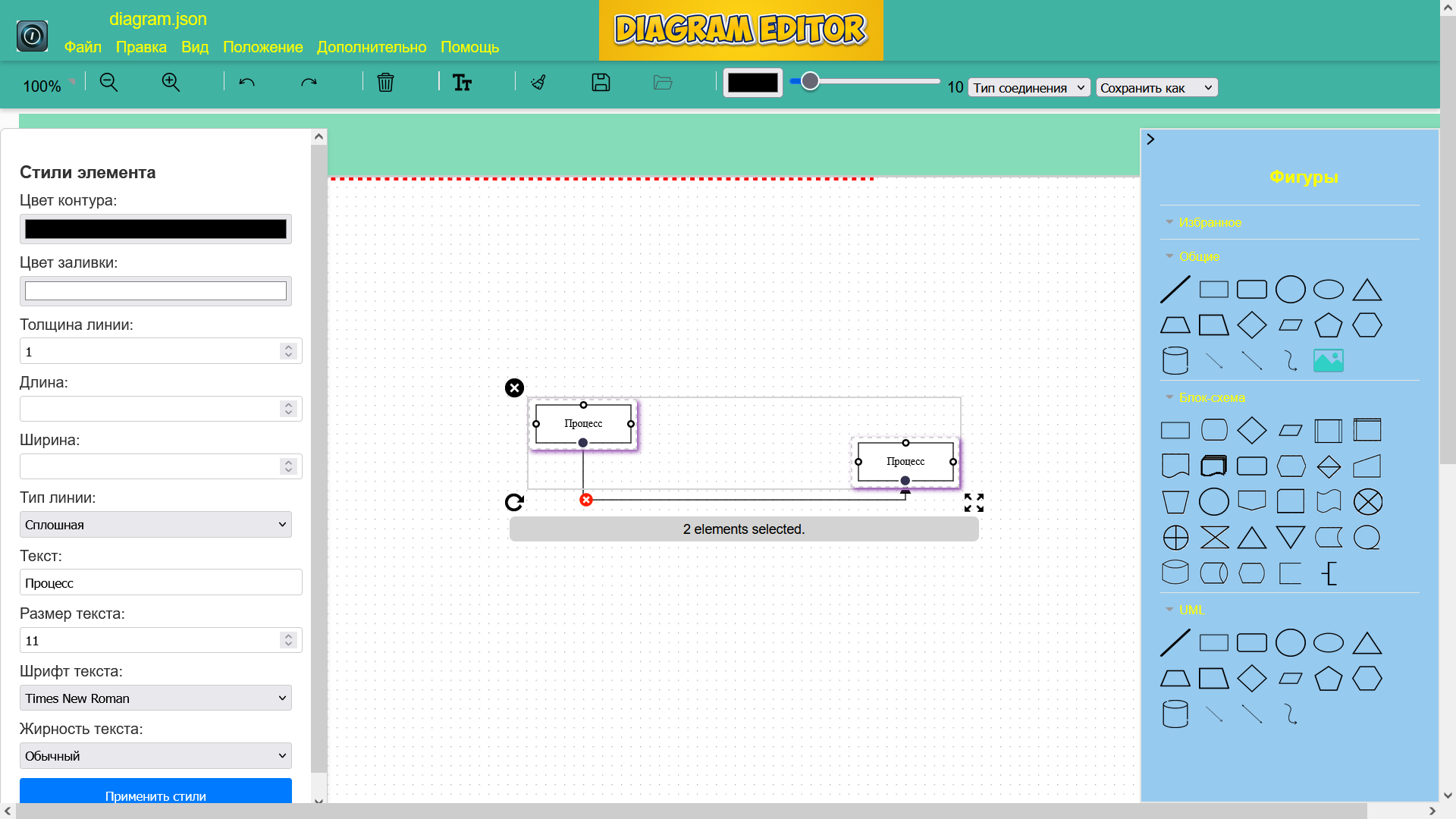


Рисунок 4.7 – Выделение фигур

Для копирования и вставки можно использовать сочетание клавиш “ctrl+c” и “ctrl+v”. На рисунке 4.8 изображено копирование и вставка фигур.

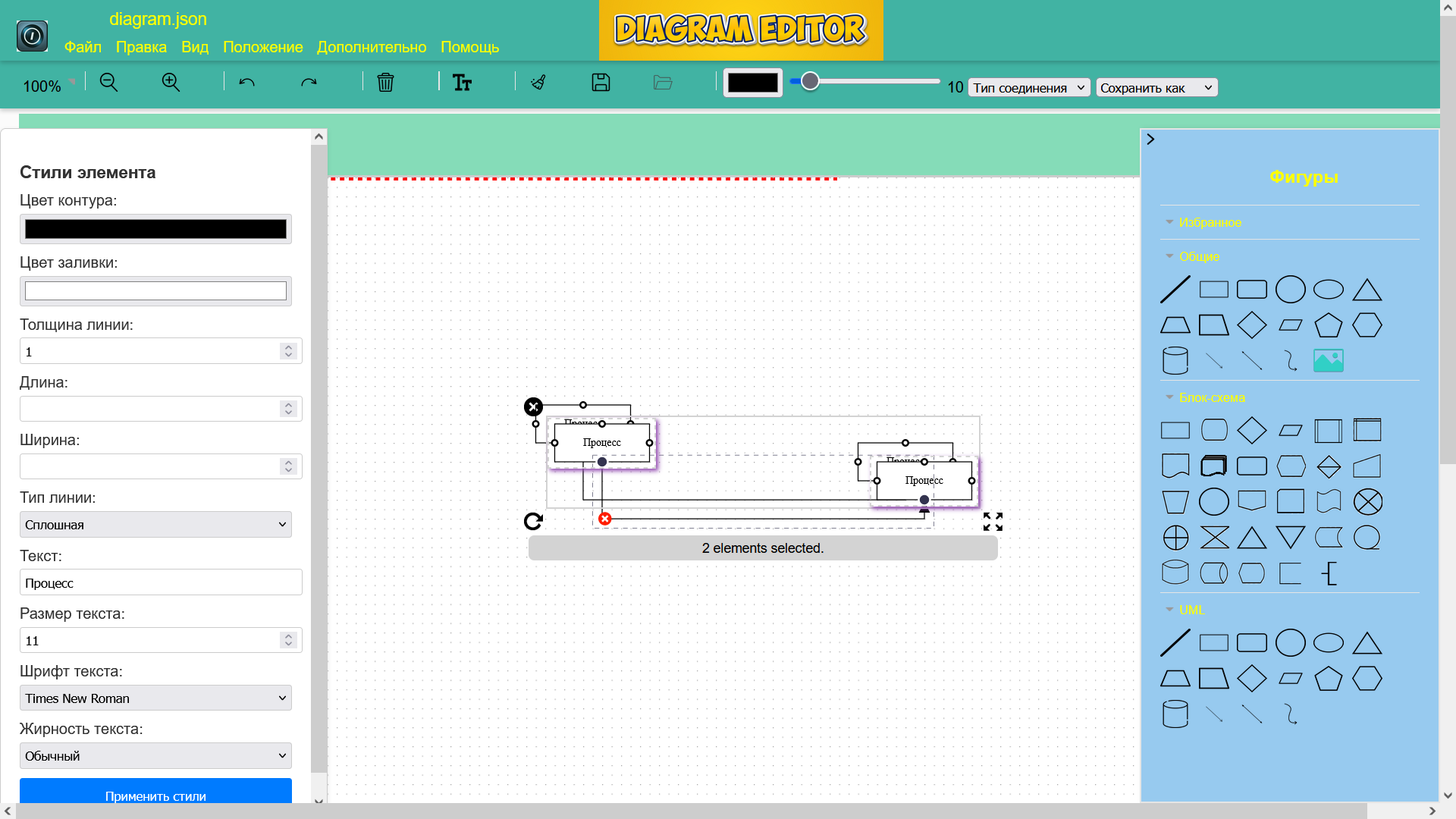


Рисунок 4.8 – Копирование и вставка фигур

Для отметы действия и повтора действия можно использовать сочетание клавиш “ctrl+z” и “ctrl+y” или соответствующие кнопки в панели инструментов. На рисунке 4.9 изображена отмена действия.

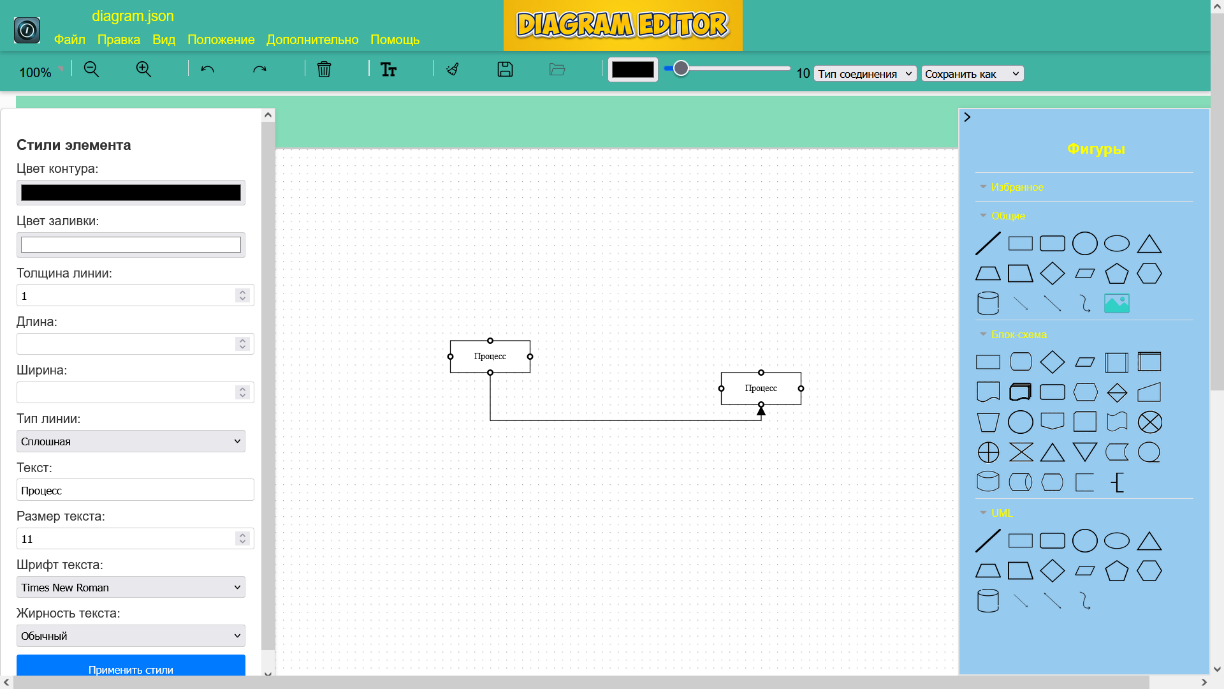


Рисунок 4.9 – Отмена действия

Чтобы сохранить диаграмму, нужно выбрать элемент из списка «сохранить как». Сохранить диаграмму можно в форматах JSON, SVG, PNG, JPEG. На рисунке 4.10 изображено сохранение в JSON формате. На рисунке 4.11 изображено сохранение в SVG формате. На рисунке 4.12 изображено сохранение в PNG формате. На рисунке 4.13 изображено сохранение в JPEG формате.

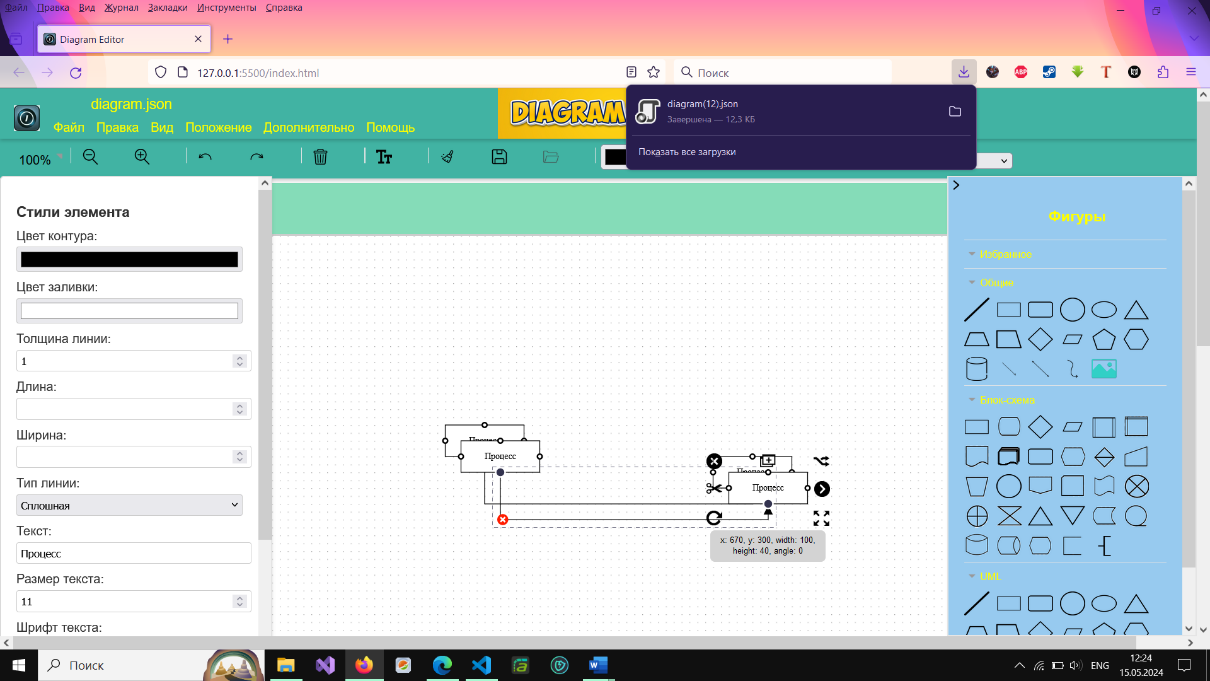


Рисунок 4.10 – Сохранение в JSON формате

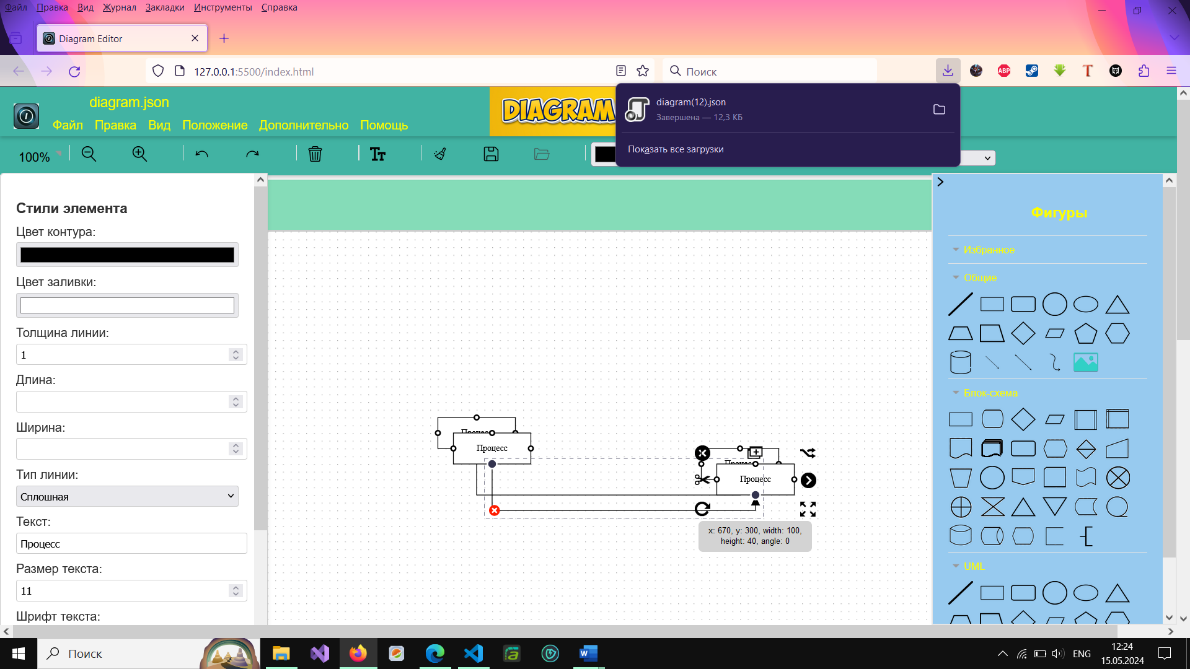


Рисунок 4.11 – Сохранение в SVG формате

Открыть диаграмму можно только в формате JSON. Для этого нужно нажать кнопку с иконкой папки на панели инструментов. На рисунке 4.14 изображено открытие диаграммы в JSON формате.

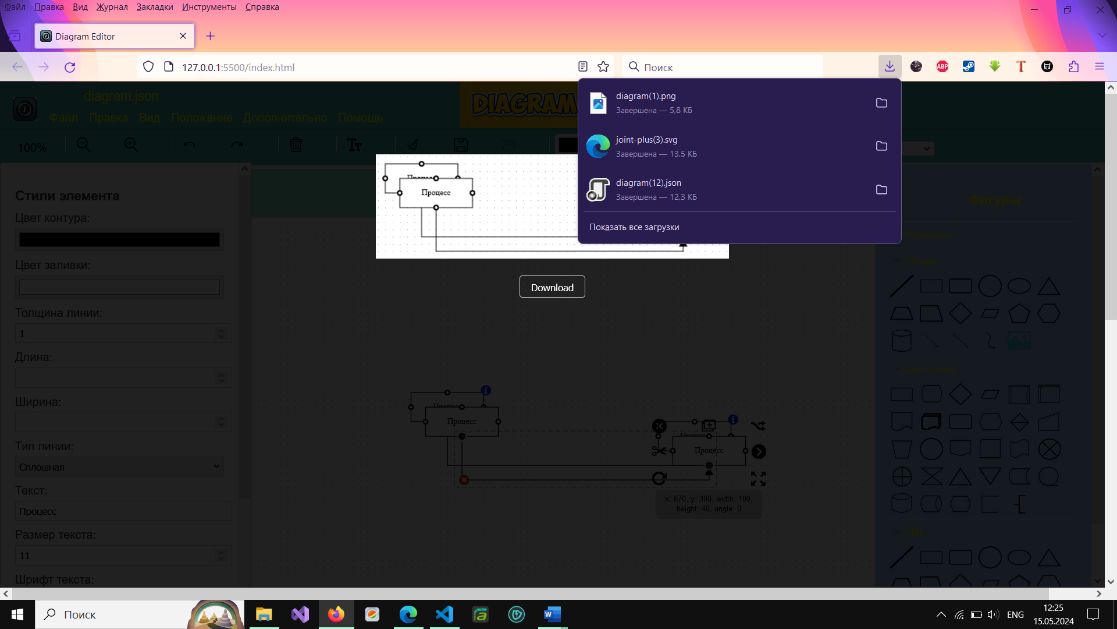


Рисунок 4.12 – Сохранение в PNG формате

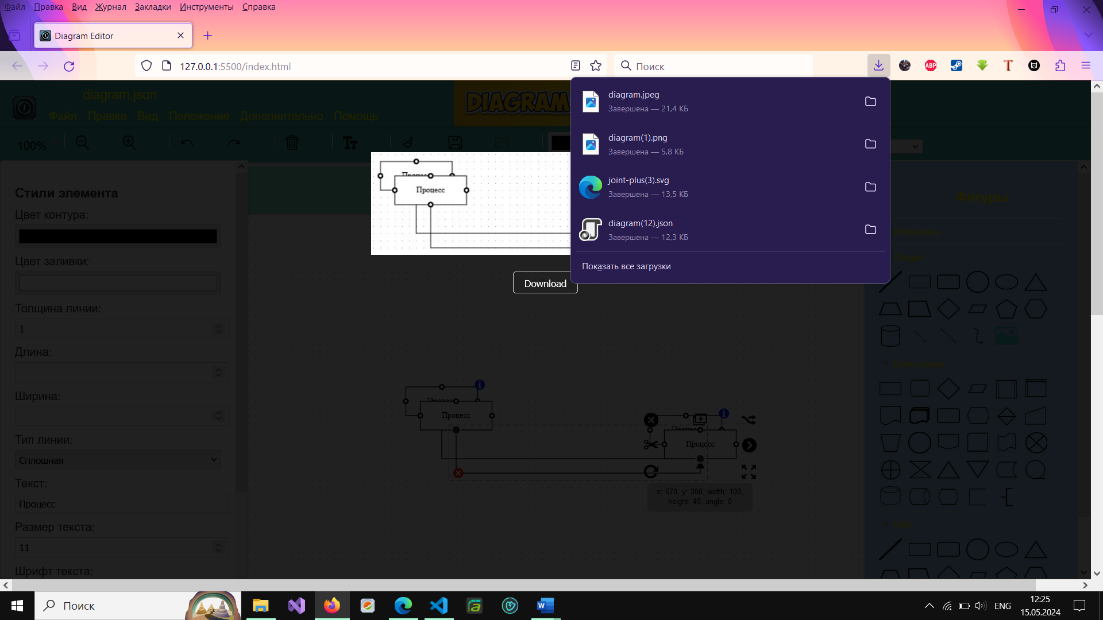


Рисунок 4.13 – Сохранение в JPEG формате

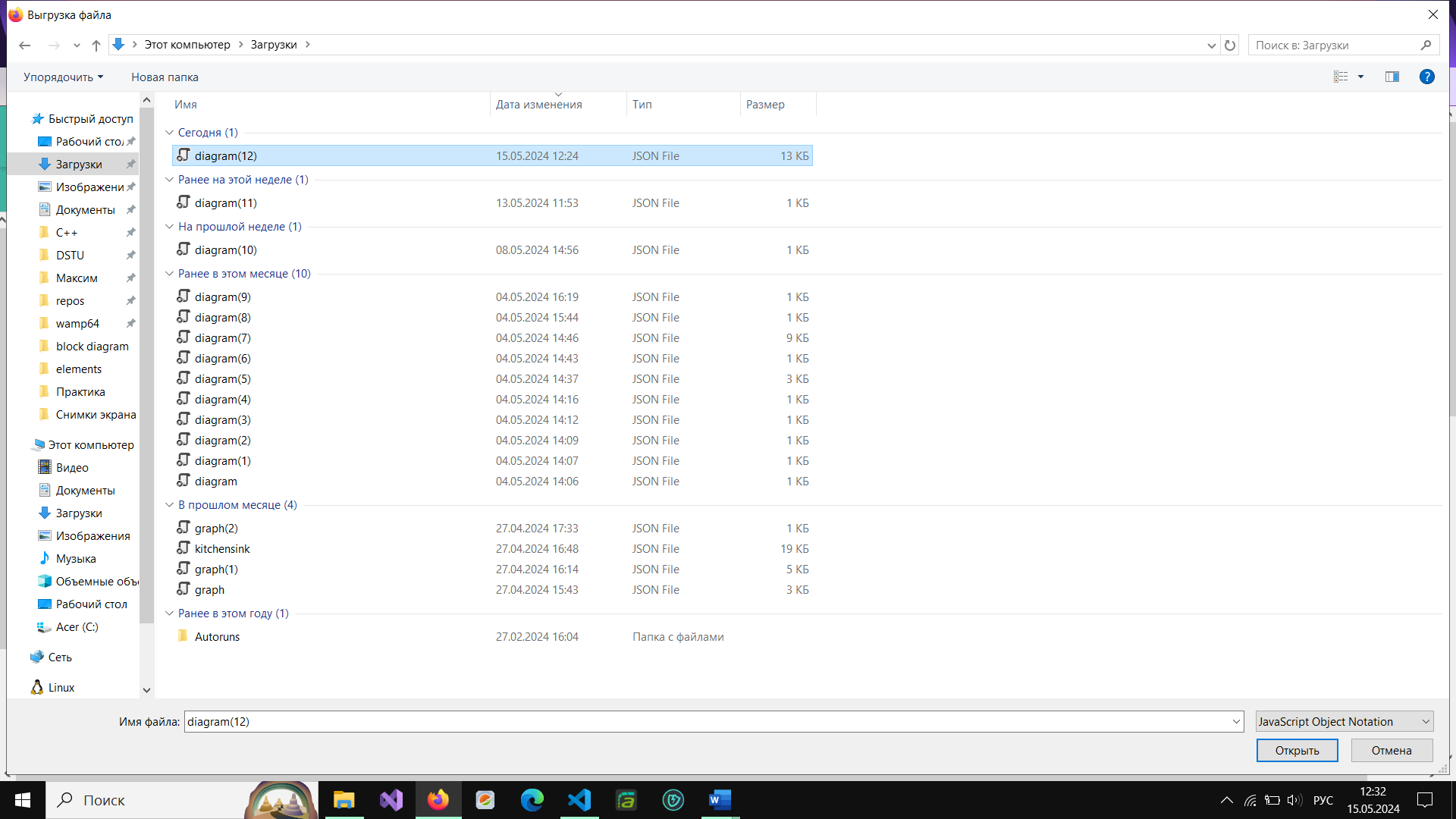


Рисунок 4.14 – Открытие диаграммы в JSON формате

## 4.2 Выводы по главе

В данной главе были продемонстрированы основные этапы тестирования векторного редактора компьютерных схем и диаграмм. Тестирование проводилось для проверки корректности работы всех реализованных функций, включая создание, перемещение, масштабирование, поворот, соединение, выделение, копирование и вставку фигур, а также отмену и повтор действий.

Для каждой функции был приведен подробный процесс тестирования, включающий пошаговое описание действий пользователя и соответствующие иллюстрации, подтверждающие корректное выполнение операций. Тестирование показало, что все функции выполняются правильно, без ошибок, обеспечивая ожидаемый результат и интуитивное взаимодействие с пользователем.

По результатам тестирования можно заключить, что все основные функции, предусмотренные целями и задачами, были реализованы в соответствии с требованиями, что делает приложение эффективным и комфортным в использовании.

5 Экономическое обоснование работы

5.1 Основные аспекты реализации работы

В данном разделе выполняются организационно-экономические расчеты, включающие в себя следующие разделы: построение маркетинговой модели, стратегический маркетинговый анализ целесообразности применения программного продукта, расчет продолжительности работ по реализации и затрат на разработку и внедрение программного продукта.

Целью разработки является создание векторного редактора для схем и диаграмм. Функциональное назначение программного средства заключается в предоставлении пользователям возможностей для создания, редактирования и хранения компьютерных схем и диаграмм в онлайн формате, с дополнительными преимуществами: интуитивно понятным интерфейсом, расширяемым набором инструментов для работы с различными фигурами, а также поддержкой сохранения и загрузки проектов в различных форматах.

Данное приложение позволяет пользователям создавать схемы и диаграммы, используя широкий набор фигур и инструментов, задавать характеристики объектов, сохранять и загружать проекты в форматах SVG, PNG, и JSON. Оно также предоставляет возможность объединять фигуры с помощью соединительных линий, масштабировать, перемещать и копировать элементы. Функционал отмены и повтора действий обеспечивает удобство при создании и редактировании схем.

Уникальность программного средства заключена в комбинации факторов, выделяющих редактор на фоне конкурентов: отсутствие рекламы, современный и стильный пользовательский интерфейс, а также поддержка технологий клиент-серверного взаимодействия, обеспечивающая совместимость с различными операционными системами и веб-браузерами.

5.2 Характеристика ПП с позиции маркетинга

Разрабатываемый программный продукт представляет собой векторный редактор для схем и диаграмм, обладающий всем функционалом традиционных инструментов для создания графических элементов, а также дополнительными возможностями, характерными для веб-приложений.

К основному функционалу относится интуитивно понятный интерфейс для добавления, редактирования и удаления элементов на схемах и диаграммах. Пользователи могут легко создавать и модифицировать графические элементы, используя набор инструментов, включающий различные фигуры и методы работы с ними, такие как соединение, масштабирование, перемещение и копирование. Система свойств объектов позволяет задавать и изменять характеристики фигур, обеспечивая гибкость и точность при создании схем.

Также данный программный продукт имеет продуманный пользовательский интерфейс, который обеспечивает комфортную работу на различных устройствах, включая компьютеры и мобильные устройства. Это позволяет пользователям всегда иметь доступ к своим проектам и работать над ними в любом месте. Современный и стильный дизайн приложения способствует удобству использования и повышает производительность работы.

Наконец, векторный редактор предлагает уникальные возможности, такие как возможность сохранения и загрузки созданных схем и диаграмм в различных форматах, таких как SVG, PNG, JPEG и JSON. Это обеспечивает совместимость с другими программами и платформами, что делает продукт более гибким и удобным в использовании.

Используя векторный редактор, пользователи смогут не только быстрее и проще создавать и редактировать схемы и диаграммы, но и улучшить свою продуктивность благодаря удобному интерфейсу и мощным инструментам. Программный продукт также поддерживает клиент-серверное взаимодействие, что позволяет работать с проектами в режиме реального времени и обеспечивает синхронизацию данных между различными устройствами пользователя.

Результаты рассмотрения характеристик ПП с позиции маркетинга представлены на рисунке 5.1.



Рисунок 5.1 – Схема маркетинговой модели программного продукта

5.3 Стратегический маркетинговый анализ целесообразности применения ПП

Для определения целесообразности применения ПП с позиций стратегического маркетинга был проведен SWOT-анализ, с помощью которого были определены 4 фактора: сильные стороны, слабые стороны, возможности и угрозы. Для определения сильных и слабых сторон организации в аспекте внедрения разработанного ПО, в таблице 5.1 были определены стратегические определяющие для SWOT-анализа.

Таблица 5.1 – SWOT-анализ программного продукта

|  |  |
| --- | --- |
| Сильные стороны | Слабые стороны |
| Интуитивно понятный интерфейс, широкий набор инструментов для создания и редактирования схем и диаграмм. | Возможные сложности в освоении для новых пользователей, отсутствие офлайн-режима |
| Высокая гибкость и кастомизация, поддержка различных форматов файлов. | Зависимость от стабильного интернет-соединения. |

5.4 Определение продолжительности работ по разработке ПП

Для определения продолжительности работ по разработке программного продукта был составлен перечь работ и была определена их продолжительность. В разработке ПП принимали участие преподаватели-консультанты, руководитель и программист.

Определение этапов разработки и их продолжительности представлены в таблице 5.2.

Таблица 5.2 – Определение этапов разработки и их продолжительности

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | Разработка технического задания (ТЗ) | Изучение предметной области | 12 | 36 | 22 | Программист |
| Разработка и утверждение ТЗ | 10 | 24 | 16 | Программист, руководитель |
| Выбор способов создания приложения «Diagram Editor» | 4 | 12 | 7 | Программист |
| 2 | Подготовительные работы | Изучение особенностей функционирования аналогичных приложений для создания схем и диаграмм и составление отчета | 8 | 20 | 13 | Программист |
| Разработка разделов, связанных с созданием веб-приложения «Diagram Edtior» | 48 | 72 | 58 | Программист |
| Проверка разделов, связанных с созданием веб-приложения «Diagram Edtior» | 6 | 12 | 8 | Руководитель |
| Изучение реализации интерфейса веб-приложения | 18 | 40 | 27 | Программист |
| Разработка макета веб-приложения «Diagram Edtior» | 24 | 36 | 32 | Программист |
| Проверка и утверждение макета веб-приложения «Diagram Edtior» | 2 | 6 | 4 | Руководитель |

Окончание таблицы 5.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 3 | Разработка  программного  обеспечения | Программирование по требованиям,  предъявленным к разработке приложения | 480 | 720 | 576 | Программист |
| Анализ и исправление ошибок в программных модулях | 60 | 120 | 84 | Программист |
| 4 | Разработка  экономического  раздела | Утверждение задания на разработку экономического раздела | 1 | 4 | 2 | Преподаватель-консультант по экономической части |
| Проведение организационно-экономических расчетов, составление отчета | 8 | 20 | 13 | Программист |
| Проверка и утверждение отчета экономического раздела | 2 | 4 | 3 | Преподаватель-консультант по экономической части |
| 5 | Разработка  экологического раздела | Утверждение задания на разработку раздела БЖД | 1 | 2 | 2 | Преподаватель-консультант по части БЖД |
| Написание раздела по БЖД | 4 | 12 | 7 | Программист |
| Проверка и утверждение раздела БЖД | 1 | 2 | 2 | Преподаватель-консультант по части БЖД |
| 6 | Утверждение ВКР | Проверка отчета ВКР | 4 | 8 | 6 | Руководитель |
| Устранение ошибок | 1 | 24 | 10 | Программист |
| Составление и утверждение отчета и документации | 4 | 10 | 6 | Программист, руководитель |
| Итого | | | | | 904 |  |

Для определения продолжительности работ по разработке программного продукта используется формула 5.1.

|  |  |
| --- | --- |
|  | (5.1) |

Где T – ожидаемая длительность работы в часах, – минимальная длительность работы, – максимальная длительность работы.

Общее время для разработки вышло равным 904 часам. Для определения суммарной деятельности этапов создания ПП был построен график Ганта, изображенный на рисунке 5.2.

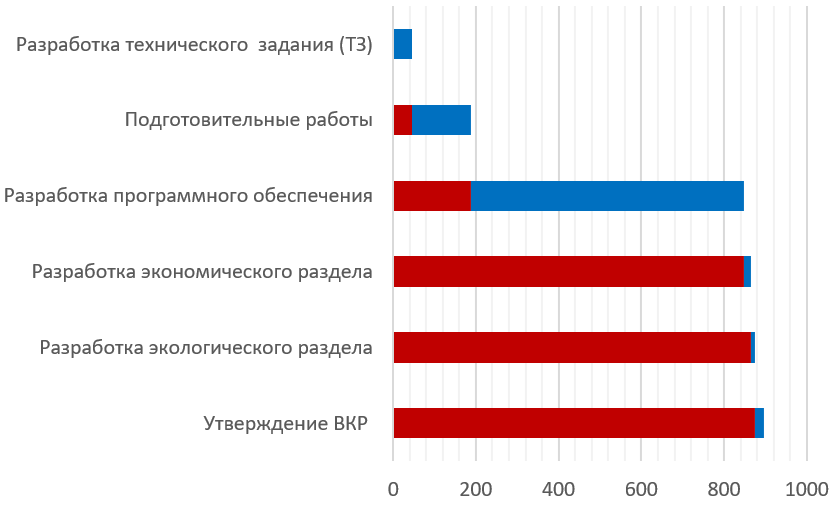


Рисунок 5.2 – График Ганта

5.5 Определение затрат на разработку и внедрение ПП

Затраты на разработку определяются по следующим статьям расходов: материальные затраты, заработная плата сотрудников, дополнительная заработная плата сотрудников, страховые взносы в государственные не бюджетные фонды, стоимость специального оборудования и программного обеспечения и накладные расходы.

Общая величина материальных затрат рассчитывается по таблице 5.3.Таблица 5.3 – Материальные затраты, связанные с созданием и внедрением ПП

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Наименование материалов | | | Ед. измерения | Количество | Цена за ед. руб. | Сумма, руб. | |
| 1 | 2 | | | 3 | 4 | 5 | 6 | |
| 1 | Бумага для печати | | | пачка | 1 | 459 | 459 | |
| 2 | Услуги связи и интернета | | | - | 4 | 500 | 2000 | |
| 3 | | Картридж для принтера | | шт. | 1 | 800 | 800 |
| 5 | | Канцелярия | | - | - | - | 500 |
|  | | | ИТОГО: | | | | 3759 |

К затратам по статье «Основная заработная плата» относятся основная заработная плата и премии. Расчет заработной платы сотрудников, используемых в разработке и внедрении программного продукта произведен в таблице 5.4.

Таблица 5.4 – Затраты по заработной плате

|  |  |  |  |
| --- | --- | --- | --- |
| Исполнитель | Часовая плата, руб. | Трудоемкость, часы | Затраты по заработной плате, руб. |
| Программист | 476 | 871 | 414596 |
| Руководитель | 297 | 40 | 11880 |
| Преподаватель-консультант по экономической части | 266 | 5 | 1330 |
| Преподаватель-консультант по части БЖД | 266 | 4 | 1064 |
| ИТОГО: | | | 428870 |

Дополнительная заработная плата () определяется в процентах от основной заработной платы программистов и руководителя. Норматив доплат принимается в размере 20% (а). Расчет представлен в формуле 5.2.

|  |  |
| --- | --- |
|  | (5.2) |
|  |  |

Страховые взносы в государственные внебюджетные фонды (СВ) с заработной платы (основной и дополнительной) определяются в соответствии с установленным законодательством РФ процентом (в 2024 году b=30%). Расчет представлен в формуле 5.3.

|  |  |
| --- | --- |
|  | (5.3) |
|  |  |

Расчет стоимости использованного оборудования и программного обеспечения произведен в таблице 5.5.

Таблица 5.5 – Стоимость использованного оборудования и программного

обеспечения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование компьютерного оборудования,  оргтехники, программного обеспечения | Количество, шт. | Цена за ед. руб. | Сумма, руб. |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Ноутбук Acer Nitro 5 | 1 | 75000 | 75000 |
| 2 | ПО Office 365 | 1 | 4399 | 4399 |
| ИТОГО: | | | | 79399 |

Расчет суммы амортизационных произведен в таблице 5.6.

Таблица 5.6 – Амортизационные отчисления основных средств производства

(основного оборудования и программного обеспечения) за год

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  ОСП | Общая первоначальная  стоимость, руб | Годовая норма  амортизации, % | Сумма годовой  амортизации, руб. |
| 1 | 2 | 3 | 4 |
| 1 | 75000 | 20 | 15000 |
| 2 | 4399 | 20 | 880 |
| ИТОГО: | | | 15880 |

Поскольку основные средства производства (оборудование и программное обеспечение) для разработки использовались в течении определенного периода времени, необходимо учесть только часть суммы годовой амортизации. Расчет процента длительности разработки происходит путем деления длительности разработки в рабочих днях () на число рабочих дней в году равное 255 (РД). Расчет процента длительности разработки произведен в формуле 5.4.

|  |  |
| --- | --- |
|  | (5.4) |
|  |  |

Результаты расчета амортизации основного оборудования и программного обеспечения в расчет на проект представлены в таблице 5.7.

Таблица 5.7 – Стоимость использованного оборудования и программного

обеспечения

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование основных средств  производства | Сумма  годовой  амортизации,  руб. | Сумма  амортизации  средств  производства, руб |
| 1 | 2 | 3 | 4 |
| 1 | Acer Ntiro 5 | 15000 | 6600 |
| 2 | ПО Office 365 | 880 | 387 |
| ИТОГО: | | | 6987 |

Накладные расходы (НР) определяются в процентах от основной заработной платы руководителя и программистов (с = 150%). Расчет накладных расходов представлен в формуле 5.5.

|  |  |
| --- | --- |
|  | (5.5) |
|  |  |

Таким образом, единовременные затраты на создание и внедрение ПП определяются по формуле 5.6.

|  |  |
| --- | --- |
|  | (5.6) |

Результаты подсчета себестоимости проекта приведены в таблице 5.8.

Таблица 5.8 – Себестоимость проекта

|  |  |  |
| --- | --- | --- |
| № | Наименование статей затрат | Сумма амортизация в расчете на проект, руб |
| 1 | 2 | 3 |
| 1 | Материальные затраты | 3759 |
| 2 | Зарплаты сотрудников | 428870 |
| 3 | Дополнительная зарплата | 85774 |
| 4 | Страховые взносы | 154393 |
| 5 | Амортизационные отчисления | 6987 |
| 6 | Накладные расходы | 643305 |
| Итого (Ед.з.): | | 1323088 |

Таким образом, по итогу всех необходимых расчетов была выявлена себестоимость проекта, итоговое значение которой составило 1323088 рублей.

Программный продукт распространяется на бесплатной основе и не требует оплаты за его использование. Проект не преследует коммерческих целей и не предполагает получение прибыли.

5.6 Выводы по главе

В данной главе была описана цель разработки и функциональное назначение приложения «Diagram Editor». Также были описаны уникальные преимущества программного продукта над конкурентами.

Кроме того, были выполнены организационно-экономические работы, включающие в себя: построение маркетинговой модели ПП, расчет продолжительности работ по разработке ПП, определения затрат на создание и и внедрение ПП.

6 Безопасность и экологичность выпускной квалификационной работы

## 6.1 Анализ вредных и опасных факторов, влияющих на здоровье программиста

Работа в офисе — это не только выполнение задач и достижение целей, но и сохранение здоровья и благополучия сотрудников. Однако, современные офисные среды часто сталкиваются с рядом негативных факторов, которые могут оказывать вредное воздействие на здоровье и самочувствие работников. Поэтому важно внимательно анализировать и контролировать эти факторы для обеспечения безопасной и здоровой рабочей среды.

Электромагнитное излучение с высокой частотой, которое часто испускается различной электронной техникой, может представлять опасность для здоровья сотрудников, особенно в современных офисных условиях. Это подтверждено нормами СанПиН 2.1.8/2.2.4.1383-03. Негативное воздействие электромагнитных полей может проявляться в виде усталости, сонливости, головных болей, а также возможны различные кожные реакции, включая зуд и жжение.

Шум в офисе является серьезной проблемой, которая привлекает значительное внимание. Нормативные требования к уровню шума в офисных помещениях установлены в ГОСТ 23337-2014, который определяет методики измерения и контроля шума в жилых и общественных зданиях. Этот шум может исходить из разнообразных источников, включая работу офисной техники, например, принтеров и копировальных аппаратов, а также общение между сотрудниками и воздействие внешних факторов, вроде уличного трафика.

Длительное воздействие шума может вызвать такие негативные воздействия как: ряд негативных последствий для здоровья и благополучия сотрудников. Включая, но не ограничиваясь: усталостью, раздражительностью, снижением способности к концентрации, ухудшением памяти и стрессовыми состояниями. Это не только влияет на эмоциональное и физическое состояние работников, но и может снижать общую эффективность работы организации.

Еще одним значимым аспектом, который привлекает наше внимание, является проблема накопления пыли в офисных помещениях. В соответствии с требованиями ГОСТа ИСО 14644-1-2017, который регламентирует уровень чистоты воздуха, следует отметить, что пыль внутри помещений может быть носителем разнообразных вредных веществ, включая угарный газ, аллергены и патогены, способные вызвать различные заболевания. Эта пыль часто скапливается в системах вентиляции, что повышает риск аллергических реакций и проблем с дыханием у работников. Более того, загрязненные воздуховоды могут стать источником различных опасных микроорганизмов, что дополнительно угрожает здоровью персонала.

Уровень освещенности на рабочих местах имеет прямое воздействие на здоровье глаз и общее самочувствие работников. Невыполнение норм освещенности СанПиН 1.2.3685-21, может привести к ухудшению зрения, раздражению глаз, повышенной утомляемости и снижению общей производительности труда.

Для уменьшения негативного воздействия этих факторов необходимо регулярно оценивать рабочую среду и принимать соответствующие меры по улучшению условий труда. Это может включать в себя использование специальных устройств для очистки воздуха, техническое обслуживание систем вентиляции, выбор удобного местоположения офиса и оптимизацию освещения. Такие меры не только способствуют улучшению здоровья и благополучия сотрудников, но и повышают их производительность и эффективность работы, что в конечном итоге благоприятно сказывается на успехе организации.

## 6.2 Устойчивость к чрезвычайным ситуациям. Пожаробезопасность

Развитие стратегий для повышения устойчивости офисного пространства к чрезвычайным ситуациям, в частности к пожарам, является неотъемлемой частью ответственного управления организацией. Понимание важности соблюдения правил пожарной безопасности в офисе становится ключевым элементом заботы как о жизнях и здоровье сотрудников, так и о сохранности имущества компании.

Правила пожарной безопасности для офисных помещений определяются специалистом по пожарной безопасности в каждой компании и утверждаются её руководством. Соблюдение этих правил существенно снижают вероятность возникновения пожаров.

Основными законодательными актами, регулирующими нормы пожарной безопасности в офисах, являются Федеральный закон № 123-ФЗ "Технический регламент о требованиях пожарной безопасности", принятый 22 июля 2008 года, а также Правила противопожарного режима Российской Федерации, утвержденные Правительством РФ 16 сентября 2020 года (постановление № 1479).

Важнейшими аспектами обеспечения пожарной безопасности в офисах являются:

* размещение табличек с номером для вызова пожарной охраны и контактами ответственных лиц по безопасности на доступных для всех сотрудников местах.
* создание и регулярное обновление планов эвакуации для каждого этажа офисного здания, а также наличие сводного плана, хранимого у ответственного лица. В случае нахождения офиса в арендованном помещении бизнес-центра требуется локальный план эвакуации.
* распространение знаков и информационных материалов по пожарной безопасности, включая стенды, плакаты и таблички, которые обучают персонал правильным действиям при возникновении пожара.
* наличие первичных средств пожаротушения, таких как огнетушители, для быстрого реагирования на начальные признаки возгорания.
* обеспечение свободного доступа к эвакуационным выходам и путям, что является критически важным для быстрой и безопасной эвакуации персонала в чрезвычайных ситуациях.

Поддержание правильного подхода к пожарной безопасности, включая проведение регулярных инструктажей, проверку состояния огнетушителей и актуализацию планов эвакуации, помогает существенно снизить риски для жизни и здоровья сотрудников, а также минимизировать материальные потери при возникновении пожара. Это способствует созданию безопасного и устойчивого рабочего пространства, обеспечивает готовность персонала к действиям в экстренных ситуациях и поддерживает стабильность организации в лице потенциальных угроз.

## 6.3 Расчет системы искусственного освещения помещений

На рисунке 6.1 представлена схема помещения с указанием его размеров и расположением окон и компьютерных столов.

Необходимо произвести расчет искусственного освещения, выполненного светильниками ЛСП01 с двумя люминесцентными лампами длинной 𝑙𝐶𝐵 = 1532 мм. Исходные данные представлены в таблице 6.1.

Определим высоту подвеса светильника над рабочей поверхностью по следующей формуле:

, (6.1)

где м, высота рабочей поверхности,

ℎ𝑐 = 0,25 (ℎ − ℎ𝑝). (6.2)

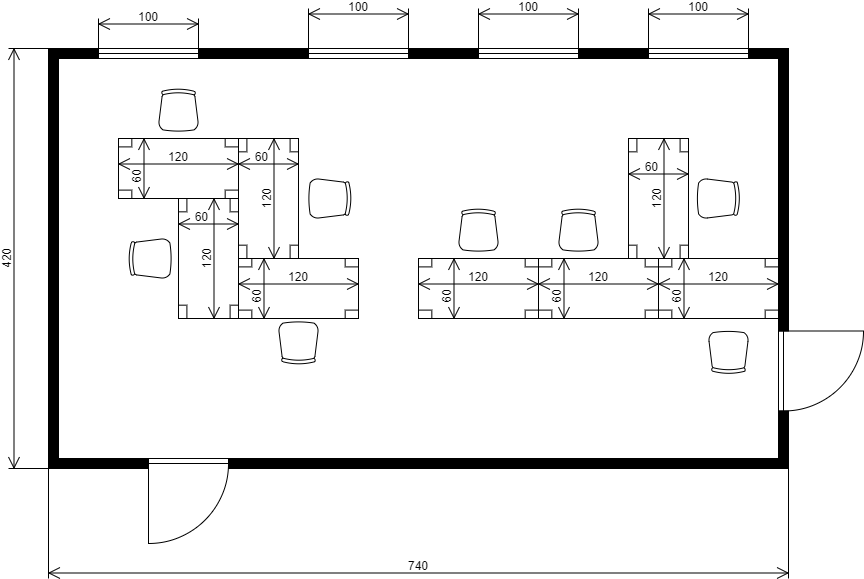


Рисунок 6.1 – Схема помещения

Таблица 6.1 – Исходные данные

|  |  |
| --- | --- |
| Наименование | Величина |
| Длина помещения, А, м | 7,4 |
| Ширина помещения, В, м | 4,2 |
| Высота помещения, h, м | 2,9 |
| Коэффициент отражения потолка | 30 |
| Коэффициент отражения стен | 10 |
| Коэффициент отражения пола | 10 |
| Нормируемая минимальная освещенность, , лк | 300 |
| Число ламп в светильнике, n, шт | 2 |
| Длина ламп, , м | 1,532 |
| Коэффициент запаса, К, | 1,5 |
| Коэффициент использования светового потока, η | 44 |
| Мощность лампы, P, Вт | 40 |
| Световой поток, , лм | 3040 |

Поскольку высота помещения , применяем потолочное крепление светильников , возьмем значение ℎ𝑐 = 0,2 м

Определим индекс помещения:

(6.3)

Используя дополнительные материалы, определим коэффициент запаса K, а также коэффициент использования светового потока 𝜂. В нашем случае эти переменные будут иметь следующие значения:

𝐾 = 1,5

𝜂 = 44

Определим расстояние между рядами светильников и от стен до ближайшего ряда по формуле:

L = λHП, (6.4)

где 𝜆 = 1,1 − коэффициент оптимального расстояния между геометрическими центрами светильников по светотехническим требованиям.

L = 1,1 1,9 = 2,09 м.

Расстояние от стен до ближайшего ряда по рекомендациям вычитывается по следующей формуле:

0,3 L = 0,3 2,09 = 0,627 м.

Далее необходимо определить общее количество светильников N при условии равномерного освещения, а также количество светильников по длине и ширине:

(6.5)

. (6.6)

(6.7)

.

Определим световой поток одной лампы по следующей формуле:

(6.8)

где Z = 1,1 – коэффициент минимальной освещенности.

Световой поток лампы должен соответствовать соотношению:

(6.9)

где – расчетный световой поток, лм;

– световой поток, лм, табличное значение;

≈

Для определения соответствия проектируемой системы освещения требованиям СП 52.13330.2016 выполним проверочный расчет по следующей формуле:

(6.10)

Так как , то освещение удовлетворяет требованиям.

Определим потребляемую мощность осветительной установки:

(6.11)

Где – мощность лампы, Вт;

– коэффициент, учитывающий потери пускорегулирующей аппаратуры. Для ЛЛ 1,25.

На рисунке 6.2 представлен эскиз размещения светильников на плане помещения.

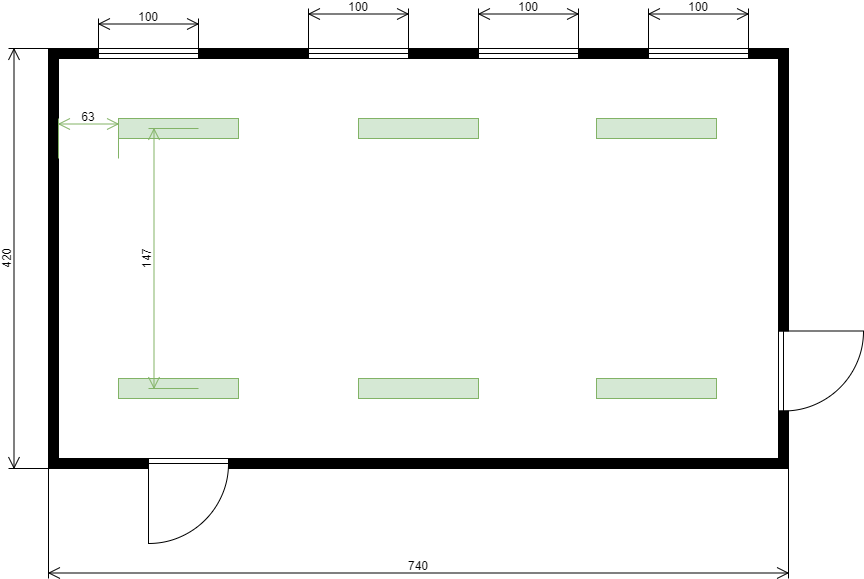


Рисунок 6.2 – Схема размещения светильников

## 6.4. Вывод по главе

В данной главе произведен расчет системы искусственного освещения рабочего помещения, а также определено, что проектируемая система освещения соответствует требованиям СП52.13330.2016.

Показаны и построены схемы размещения светильников в помещении с указанием расстояния от стен до ближайшего ряда по длине и оптимального расстояния между геометрическими центрами светильников по ширине между рядами светильников.

Также в данной главе мы описали возможные мероприятия для сокращения пагубного влияния на здоровье сотрудников офиса.

**Заключение**

В ходе выполнения данной работы было разработано веб-приложение «Diagram Editor», которое упрощает создание компьютерных схем и диаграмм. Приложение отвечает всем установленным требованиям.

Данное программное средство полностью реализует задачи, поставленные в пункте 1.6, а именно:

* интуитивно понятный интерфейс для добавления, редактирования и удаления элементов на схемах и диаграммах, обеспечивающий простоту использования как для новичков, так и для опытных пользователей;
* современный и стильный дизайн приложения, который не только улучшает визуальное восприятие, но и способствует удобству работы;
* набор инструментов, содержащих различные фигуры и методы работы с ними, включая возможность соединения, масштабирования, поворота и выравнивания элементов;
* система свойств объектов, которая позволяет определять и изменять характеристики фигур, схем и диаграмм, обеспечивая высокую гибкость и точность при создании сложных структур;
* возможность сохранения и загрузки созданных схем и диаграмм в различных форматах, таких как JSON, SVG, PNG, JPEG, что обеспечивает совместимость и удобство обмена данными.

Получены и закреплены навыки разработки и создания программного средства, связанные с полновесной разработкой приложения, реализующего клиент-серверное взаимодействие в браузерной среде.

**Перечень использованных информационных ресурсов**

1. Схемы и диграммы. Отличия схем от диаграмм [Электронный ресурс], редактировано 18 марта 2024 г.

URL: https://techwriters.ru/ru/tools\_and\_technologies/schemes\_and\_diagrams.html (дата обращения: 18.03.2024).

1. Векторная графика [Электронный ресурс], редактировано 18 марта 2024 г.

URL: [https://nauchniestati.ru/spravka/vektornoe‑predstavlenie‑izobrazhenij/?ysclid=lvjjn6nscd746971435](https://nauchniestati.ru/spravka/vektornoepredstavlenieizobrazhenij/?ysclid=lvjjn6nscd746971435) (дата обращения: 18.03.2024).

1. История появления [Электронный ресурс], редактировано 19 марта 2024 г.

URL: https://www.timetoast.com/timelines/8a02d899‑7e3e‑41b4‑8b71‑253825367e6d (дата обращения: 19.03.2024).

1. Недостатки и преимущества Diagrams.net [Электронный ресурс], редактировано 20 марта 2024 г.

URL: https://softwarekeep.com/nl‑nl/blogs/comparisons/visio‑vs‑diagrams‑net?\_\_ya\_mt\_enable\_static\_translations=1 (дата обращения: 20.03.2024).

1. Недостатки и преимущества LucidChart [Электронный ресурс], редактировано 22 марта 2024 г.

URL: <https://junior3d.ru/article/lucidchart.html?ysclid=lvjjepcdkg260121195> (дата обращения: 22.03.2024).

1. Недостатки и преимущества Microsoft Visio [Электронный ресурс], редактировано 22 марта 2024 г.

URL: https://studfile.net/preview/4365534/page:3/ (дата обращения: 22.03.2024).

1. Недостатки и преимущества SmartDraw [Электронный ресурс], редактировано 23 марта 2024 г.

URL: <https://junior3d.ru/article/smartdraw.html?ysclid=lvjjg4fiuj584199923> (дата обращения: 23.03.2024).

1. Недостатки и преимущества yEd Graph Editor [Электронный ресурс], редактировано 23 марта 2024 г.
2. JavaScript [Электронный ресурс], редактировано 23 марта 2024 г.

URL: https://ru.wikipedia.org/wiki/JavaScript (дата обращения: 23.03.2024).

1. VS Code [Электронный ресурс], редактировано 26 марта 2024 г.

URL: https://ru.wikipedia.org/wiki/Visual\_Studio\_Code (дата обращения: 26.03.2024).

1. Git [Электронный ресурс], редактировано 26 марта 2024 г.

URL: https://git-scm.com/ (дата обращения: 26.03.2024).

1. JointJS [Электронный ресурс], редактировано 26 марта 2024 г.

URL: https://products.fileformat.com/ru/diagram/javascript/joint/ (дата обращения: 26.03.2024).

1. SVG [Электронный ресурс], редактировано 27 марта 2024 г.

URL: https://developer.mozilla.org/ru/docs/Web/SVG (дата обращения: 27.03.2024).

**Приложение** **А** **Техническое задание**

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Декан фак. «ИиВТ»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Поркшеян В.М.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. | УТВЕРЖДЕНО  Зав. каф. «ПОВТ и АС»  \_\_\_\_\_\_\_\_\_\_\_\_\_В. В. Долгов  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

**А.1 Общие сведения**

**А 1.1 Наименование программного средства**

Полное наименование программного средства: «Векторный редактор компьютерных схем и диаграмм», краткое наименование: «Diagram Editor».

**А 1.2 Область применения программного средства**

Областью применения данного программного средства является сфера образования.

**А.2 Основания для разработки**

Разработка ведется на основании документа «Учебный план для студентов ВУЗа», факультета «Информатики и вычислительной техники» (ИиВТ) направление подготовки 09.03.04 – «Программная инженерия» Донского Государственного Технического Университета (ДГТУ).

**А.3 Назначение разработки**

**А.3.1 Функциональное назначение**

Функциональное назначение программного средства – создание и визуализация компьютерных схем и диаграмм.

**А.3.2 Эксплуатационное назначение**

Эксплуатационное назначение программного средства заключается в его использовании на ПК под управлением операционной системы Windows 10 и более, macOS или Linux.

**А.4 Требования к программе**

**А.4.1 Требования к функциональным характеристикам**

Программное средство должно осуществлять следующие функции:

* создание фигур и добавление их на холст;
* перемещение фигур;
* масштабирование фигур;
* вращение фигур;
* соединение фигур;
* редактирование свойств каждой фигуры;
* создание диаграммы с определенным названием;
* сохранение и открытие диаграмм в форматах json, svg, jpeg, png;
* возможность работать с несколькими экземплярами диаграмм;
* выделение фигур;
* копирование фигур;
* вставка фигур;
* удаление фигур;
* отмена действия;
* редактирование текста внутри фигур;
* внешнее масштабирование холста (приближение и отдаление);
* внутреннее масштабирование холста (изменение размеров);
* изменение типа соединения фигур;
* изменение размера и цвета сетки.

**А.4.2 Требования к надежности**

Надежное функционирование программного комплекса должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

* организация бесперебойного питания технических средств;
* организация правильного использования модулей;
* защиты программного средства от несанкционированного проникновения;
* копирование (архивирование) данных, необходимых для восстановления работы программного средства;
* регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

**А.4.2.1 Входные данные**

Входными данными для программного средства являются действия пользователя.

**А.4.2.2 Выходные данные**

Выходными данными являются графические элементы, которые отображаются на экране.

**А.4.3 Условия эксплуатации**

Для функционирования программного продукта необходимо соблюдение всех требований и правил эксплуатации компьютерной техники. Для работы с программным средством необходимы навыки работы с ПК под управлением операционной системы Windows 10 версии и более, macOS или Linux на уровне пользователя. Дополнительных требований и ограничений не вводится.

**А.4.4 Требования к составу и параметрам технических средств**

Для эксплуатации программного средства требуется персональный компьютер с установленной операционной системой Windows версии 10, macOS или Linux, наличие 4 гБ оперативной памяти, 64-битный процессор, интернет-соединение не менее 128 кб/c.

**А.4.5 Требования к исходным кодам и языкам программирования**

Клиентская часть должна быть реализована с использованием языка программирования JavaScript. Для создания интерфейса программного средства должен использоваться язык разметки HTML, а для реализации клиентской части программного средства необходимо использовать язык программирования JavaScript.

На стороне клиента требуется наличие современного веб-браузера, поддерживающего последние стандарты HTML5, CSS3 и ECMAScript 6 (ES6).

Исходный код должен быть хорошо структурирован и документирован, следовать принципам модульности и обеспечения легкости сопровождения и расширения.

**А.4.6 Требования к транспортировке и хранению**

Условия транспортирования, места хранения, условия складирования и сроки хранения в различных условиях должны соответствовать требованиям, предъявляемым к носителям информации, на которых будет содержаться данное программное изделие.

Допустимы все способы транспортирования и хранения, не нарушающие целостность используемого носителя данных. Программное средство может храниться на любом носителе информации, имеющее возможность подключения к персональному компьютеру.

**А.4.7 Специальные требования**

Оценочные требования к функционалу программного средства и уровню его реализации сведены в таблицу А.1.

Таблица А.1 - Степень самостоятельности разработки программного средства

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Реализуемые программным средством функции | Степень самостоятельности разработки  (предусмотрено 3 уровня:  «новое», «модиф.», «заимств.» | |
| Алгоритм | ПС |
| 1 | создание фигур и добавление их на холст | новое | новое |
| 2 | перемещение фигур | заимств. | новое |
| 3 | масштабирование фигур | заимств. | новое |
| 4 | вращение фигур | заимств. | новое |
| 5 | соединение фигур | заимств. | новое |
| 6 | редактирование свойств каждой фигуры | новое | новое |
| 7 | сохранение и открытие диаграмм в форматах json, png, svg, jpeg | новое | новое |
| 8 | выделение фигур | заимств. | новое |
| 9 | копирование фигур | заимств. | новое |
| 10 | вставка фигур | заимств. | новое |
| 11 | удаление фигур | новое | новое |
| 12 | отмена действия | заимств. | новое |
| 13 | редактирование текста внутри фигур | новое | новое |
| 14 | внешнее масштабирование холста | новое | новое |
| 15 | внутреннее масштабирование холста | новое | новое |
| 16 | изменение типа соединения фигур | новое | новое |
| 17 | изменение размера и цвета сетки | новое | новое |

А.5 Требования к программной документации

Программная документация должна состоять из следующих листов:

* титульный лист;
* задание на выпускную квалификационную работу;
* пояснительная записка к выпускной квалификационной работе;
* техническое задание по ГОСТ 19.201-78 ЕСПД;
* исходный код программного средства по ГОСТ 19.401-79 ЕСПД.

**А.6 Стадии и этапы разработки**

* постановка задачи (с 16.02.2024 по 21.02.2024);
* изучение предметной области (с 23.02.2024 по 18.03.2024);
* разработка алгоритмов решения задачи (с 19.03.2024 по 14.04.2024);
* разработка программы (с 15.04.2024 по 30.04.2024);
* тестирование программы (с 01.05.2024 по 04.05.2024);
* разработка пояснительной записки к данному программному средству (с 05.05.2024 по 19.05.2024).

**А.7 Порядок контроля и приемки**

Порядок и контроль приемки определяются заведующим кафедрой «ПОВТ и АС» и основаны на демонстрации знаний технологии и умении создавать программные средства для различных предметных областей.

Главным требованием к приемке является наличие правильно работающего комплекса модулей с тестовым примером и отчета, представленного в печатном виде.

|  |  |
| --- | --- |
| Разработчик технического задания:  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024. | /Лапардин Максим Леонидович /  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Приложение Б** **Листинг программы**

Исходные коды программных модулей в полном объеме представлены на прилагаемом носителе.

Листинг Б.1 – Подпрограмма index.html.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Diagram Editor</title>

<link rel="stylesheet" href="../css/properties-panel.css">

<link rel="stylesheet" href="../css/sidebar.css">

<link rel="stylesheet" href="../css/menubar.css">

<link rel="stylesheet" href="../css/toolbar.css">

<link rel="stylesheet" href="../css/style.css">

<link rel="stylesheet" href="https://unicons.iconscout.com/release/v4.0.0/css/line.css">

<link rel="stylesheet" type="text/css" href="../node\_modules/@joint/plus/joint-plus.css">

</head>

<body>

<div class="geHeaderContainer">

<div class="geMenubarContainer">

<a class="geIcon"></a>

<div class="geTextDiv">

<a id="title" class="geItem"></a>

</div>

<img src="../icons/header.png">

<div class="geMenubar">

<a class="geItem">Файл

<div id="file" class="geDropdown">

<div class="geDropdownItem" id="open-option">Открыть</div>

<div class="geDropdownItem" id="save-option">Сохранить</div>

<div class="geDropdownItem" id="exit-option">Выход</div>

</div>

</a>

<a class="geItem">Правка

<div class="geDropdown">

<div id="undo" class="geDropdownItem">Отменить</div>

<div id="redo" class="geDropdownItem">Повторить</div>

<div id="cut" class="geDropdownItem">Вырезать</div>

<div id="copy" class="geDropdownItem">Копировать</div>

<div id="paste" class="geDropdownItem">Вставить</div>

<div id="delete" class="geDropdownItem">Удалить</div>

</div>

</a>

<a class="geItem">Вид

<div class="geDropdown">

<div id="zoom" class="geDropdownItem">Увеличить</div>

<div id="zoom out" class="geDropdownItem">Уменьшить</div>

</div>

</a>

<a class="geItem">Положение

<div class="geDropdown">

<div class="geDropdownItem">Вверх</div>

<div class="geDropdownItem">Вниз</div>

<div class="geDropdownItem">Влево</div>

<div class="geDropdownItem">Вправо</div>

</div>

</a>

<a class="geItem">Дополнительно

<div class="geDropdown">

<div class="geDropdownItem">Настройки</div>

<div class="geDropdownItem">Справка</div>

</div>

</a>

<a class="geItem">Помощь

<div class="geDropdown">

<div class="geDropdownItem">О программе</div>

<div class="geDropdownItem">Поддержка</div>

</div>

</a>

</div>

</div>

<div class="geToolbarContainer">

<div class="geToolbar">

<a id="zoom-scale" class="geLabel" title="Масштаб (Alt+Mousewheel)">

100%

<img src="../icons/arrow.gif">

<div id="scale-dropdown">

<a class="scale-option" data-scale="0.25">25%</a>

<a class="scale-option" data-scale="0.5">50%</a>

<a class="scale-option" data-scale="0.75">75%</a>

<a class="scale-option" data-scale="1">100%</a>

<a class="scale-option" data-scale="1.25">125%</a>

<a class="scale-option" data-scale="1.5">150%</a>

<a class="scale-option" data-scale="1.75">175%</a>

</div>

</a>

<div class="geSeparator"></div>

<a id="zoom-button" class="geLabel" title="Увеличить">

<img src="../icons/glass1.png"></img>

</a>

<a id="unzoom-button" class="geLabel" title="Уменьшить">

<img src="../icons/glass2.png"></img>

</a>

<div class="geSeparator"></div>

<a id='undo-button'class="geLabel" title="Отменить (Ctrl+Z)">

<img src="../icons/undo.png"></img>

</a>

<a id='redo-button' class="geLabel" title="Вернуть (Ctrl+Y)">

<img src="../icons/revert.png"></img>

</a>

<div class="geSeparator"></div>

<a id='delete-button' class="geLabel" title="Удалить (Delete)">

<img src="../icons/delete.png"></img>

</a>

<div class="geSeparator"></div>

<input type="color" id="color-pick-input" title="Цвет сетки" value="#ffffff">

<input type="range" id="line-width-input" title="Размер сетки" min="1" max="100" value="20">

<span id="range-value"></span>

<a id="textBox" class="geLabel" title="Добавить текст">

<img src="../icons/text.png"></img>

</a>

<div class="geSeparator"></div>

<a id="clear-button" class="geLabel" title="очистить">

<img src="../icons/clear.png"></img>

</a>

<a id="save-button" class="geLabel" title="сохранить">

<img src="../icons/save.png"></img>

</a>

<a id="open-button" class="geLabel" title="открыть">

<img src="../icons/open.png"></img>

</a>

<div class="geSeparator"></div>

<select id="line-connection-type">

<option value="">Тип соединения</option>

<option value="manhattan">Manhattan</option>

<option value="metro">Metro</option>

<option value="normal">Normal</option>

<option value="orthogonal">Orthogonal</option>

<option value="rightAngle">Orthogonal Router with Given Direction (Right Angle)</option>

</select>

<select id="saveSelect">

<option value="">Сохранить как</option>

<option value="json">JSON</option>

<option value="svg">SVG</option>

<option value="png">PNG</option>

<option value="jpg">JPG</option>

</select>

</div>

</div>

</div>

<div id="sidebar">

<img id="btn-toggle-menu" src="../icons/next.png"></img>

<h3>Фигуры</h3>

<a class="geTitle" >Избранное</a>

<a class="geTitle" >Общие

</a>

<div class="geSidebar">

<a class="geItem" title="прямая линия" data-shape="line">

<img src="../mainFigures/line.png" ></img>

</a>

<a class="geItem" title="прямоугольник" data-shape="rectangle">

<img src="../mainFigures/rectangle.png" ></img>

</a>

<a class="geItem" title="закругленный прямоугольник" data-shape="rounded-rectangle">

<img src="../mainFigures/rounded-rectangle.png" ></img>

</a>

<a class="geItem" title="круг" data-shape="circle">

<img src="../mainFigures/circle.png" ></img>

</a>

<a class="geItem" title="эллипс" data-shape="ellipse">

<img src="../mainFigures/ellipse.png" ></img>

</a>

<a class="geItem" title="треугольник" data-shape="triangle">

<img src="../mainFigures/triangle.png" ></img>

</a>

<a class="geItem" title="трапеция" data-shape="trapezium">

<img src="../mainFigures/trapezium.png" ></img>

</a>

<a class="geItem" title="комментарий" data-shape="90trapezium">

<img src="../mainFigures/90trapezium.png" ></img>

</a>

<a class="geItem" title="ромб" data-shape="rhomb">

<img src="../mainFigures/rhomb.png" ></img>

</a>

<a class="geItem" title="параллелограмм" data-shape="parallelogram">

<img src="../mainFigures/parallelogram.png" ></img>

</a>

<a class="geItem" title="пятиугольник" data-shape="pentagon">

<img src="../mainFigures/pentagon.png" ></img>

</a>

<a class="geItem" title="шестиугольник" data-shape="hexagon">

<img src="../mainFigures/hexagon.png" ></img>

</a>

<a class="geItem" title="цилиндр" data-shape="cylinder">

<img src="../mainFigures/cylinder.png" ></img>

</a>

<a class="geItem" title="односторонняя стрелка" data-shape="arrow">

<img src="../mainFigures/arrows.png" ></img>

</a>

<a class="geItem" title="двусторонняя стрелка" data-shape="double-arrow">

<img src="../mainFigures/double-arrow.png" ></img>

</a>

<a class="geItem" title="криволинейная стрелка" data-shape="curved-arrow" >

<img src="../mainFigures/curve-arrow.png" ></img>

</a>

<a class="geItem" title="изображение" data-shape="image" >

<img src="/icons/image-icon1.svg" ></img>

</a>

</div>

<a class="geTitle" >Блок-схема</a>

<div class="geSidebar">

<a class="geItem" title="процесс" data-shape="process">

<img src="../mainFigures/rectangle.png" ></img>

</a>

<a class="geItem" title="альтернативный процесс" data-shape="alternative process">

<img src="../block diagram/alternative process.png" ></img>

</a>

<a class="geItem" title="решение/условие" data-shape="decision/condition">

<img src="../mainFigures/rhomb.png" ></img>

</a>

<a class="geItem" title="вход/выход" data-shape="inOutput">

<img src="../mainFigures/parallelogram.png" ></img>

</a>

<a class="geItem" title="определенный процесс" data-shape="predefined process">

<img src="../block diagram/predefined process.png" ></img>

</a>

<a class="geItem" title="эллипс" data-shape="internal storage">

<img src="../block diagram/internal data.png" ></img>

</a>

<a class="geItem" title="документ" data-shape="document">

<img src="../block diagram/document.png" ></img>

</a>

<a class="geItem" title="мульти-документ" data-shape="multi-document">

<img src="../block diagram/multi-document.png" ></img>

</a>

<a class="geItem" title="начало/конец" data-shape="start/end">

<img src="../mainFigures/rounded-rectangle.png" ></img>

</a>

<a class="geItem" title="подготовка" data-shape="preparation">

<img src="../block diagram/preparation.png" ></img>

</a>

<a class="geItem" title="сортировка" data-shape="sort">

<img src="../block diagram/sort.png" ></img>

</a>

<a class="geItem" title="ручной ввод" data-shape="manual input">

<img src="../block diagram/manual input.png" ></img>

</a>

<a class="geItem" title="ручная операция" data-shape="manual operation">

<img src="../block diagram/manual operation.png" ></img>

</a>

<a class="geItem" title="узел" data-shape="node">

<img src="../mainFigures/circle.png" ></img>

</a>

<a class="geItem" title="ссылка" data-shape="link">

<img src="../block diagram/link.png" ></img>

</a>

<a class="geItem" title="карточка" data-shape="card">

<img src="../block diagram/card.png" ></img>

</a>

<a class="geItem" title="перфолента" data-shape="tape">

<img src="../block diagram/tape.png" ></img>

</a>

<a class="geItem" title="узел суммирования" data-shape="summing junction">

<img src="../block diagram/summing junction.png" ></img>

</a>

<a class="geItem" title="узел или" data-shape="or" >

<img src="../block diagram/or.png" ></img>

</a>

<a class="geItem" title="сопоставление" data-shape="collate" >

<img src="../block diagram/collate.png" ></img>

</a>

<a class="geItem" title="извлечение" data-shape="exctract/measurement" >

<img src="../mainFigures/triangle.png" ></img>

</a>

<a class="geItem" title="объединение" data-shape="merge/storage" >

<img src="../mainFigures/triangle.png" style="transform: rotate(180deg);" ></img>

</a>

<a class="geItem" title="сохраненные данные" data-shape="stored data" >

<img src="../block diagram/stored data.png" ></img>

</a>

<a class="geItem" title="память с последовательным доступом" data-shape="sequantional data" >

<img src="../block diagram/sequantional data.png" ></img>

</a>

<a class="geItem" title="база данных" data-shape="database" >

<img src="../block diagram/database.png" ></img>

</a>

<a class="geItem" title="память с прямым доступом" data-shape="direct data" >

<img src="../block diagram/direct data.png" ></img>

</a>

<a class="geItem" title="дисплей" data-shape="display" >

<img src="../block diagram/display.png" ></img>

</a>

<a class="geItem" title="annotation1" data-shape="annotation1" >

<img src="../block diagram/annotation1.png" ></img>

</a>

<a class="geItem" title="комментарий2" data-shape="annotation2" >

<img src="../block diagram/annotation2.png" ></img>

</a>

</div>

<a class="geTitle" >UML</a>

<div class="geSidebar">

<a class="geItem" title="класс" data-shape="class">

<img src="../mainFigures/line.png" ></img>

</a>

<a class="geItem" title="интерфейс" data-shape="interface">

<img src="../mainFigures/rectangle.png" ></img>

</a>

<a class="geItem" title="обьект" data-shape="rounded-rectangle">

<img src="../mainFigures/rounded-rectangle.png" ></img>

</a>

<a class="geItem" title="круг" data-shape="circle">

<img src="../mainFigures/circle.png" ></img>

</a>

<a class="geItem" title="эллипс" data-shape="ellipse">

<img src="../mainFigures/ellipse.png" ></img>

</a>

<a class="geItem" title="треугольник" data-shape="triangle">

<img src="../mainFigures/triangle.png" ></img>

</a>

<a class="geItem" title="трапеция" data-shape="trapezium">

<img src="../mainFigures/trapezium.png" ></img>

</a>

<a class="geItem" title="комментарий" data-shape="90trapezium">

<img src="../mainFigures/90trapezium.png" ></img>

</a>

<a class="geItem" title="ромб" data-shape="rhomb">

<img src="../mainFigures/rhomb.png" ></img>

</a>

<a class="geItem" title="параллелограмм" data-shape="parallelogram">

<img src="../mainFigures/parallelogram.png" ></img>

</a>

<a class="geItem" title="пятиугольник" data-shape="pentagon">

<img src="../mainFigures/pentagon.png" ></img>

</a>

<a class="geItem" title="шестиугольник" data-shape="hexagon">

<img src="../mainFigures/hexagon.png" ></img>

</a>

<a class="geItem" title="цилиндр" data-shape="cylinder">

<img src="../mainFigures/cylinder.png" ></img>

</a>

<a class="geItem" title="односторонняя стрелка" data-shape="arrow">

<img src="../mainFigures/arrows.png" ></img>

</a>

<a class="geItem" title="двусторонняя стрелка" data-shape="double-arrow">

<img src="../mainFigures/double-arrow.png" ></img>

</a>

<a class="geItem" title="криволинейная стрелка" data-shape="curved-arrow" >

<img src="../mainFigures/curve-arrow.png" ></img>

</a>

</div>

</div>

<div id="diagram-container">

<div id="canvas" >

</div>

<div id="radar" >

</div>

</div>

<div id="properties-panel">

<h2>Стили элемента</h2>

<label for="stroke-color">Цвет контура:</label>

<input type="color" id="stroke-color"><br>

<label for="fill-color">Цвет заливки:</label>

<input type="color" id="fill-color"><br>

<label for="line-width">Толщина линии:</label>

<input type="number" id="line-width" min="1" max="10" value="1"><br>

<label for="width">Длина:</label>

<input type="number" id="width" min="1" max="1000" ><br>

<label for="height">Ширина:</label>

<input type="number" id="height" min="1" max="1000" ><br>

<label for="line-style">Тип линии:</label>

<select id="line-style">

<option value="solid">Сплошная</option>

<option value="dashed">Пунктирная</option>

<option value="dotted">Точечная</option>

</select><br>

<label for="text-content">Текст:</label>

<input type="text" id="text-content"><br>

<label for="text-size">Размер текста:</label>

<input type="number" id="text-size" min="8" max="24" value="12"><br>

<label for="text-font">Шрифт текста:</label>

<select id="text-font">

<option value="Arial">Arial</option>

<option value="Helvetica">Helvetica</option>

<option value="Times New Roman">Times New Roman</option>

<!-- Добавьте другие варианты шрифтов по вашему усмотрению -->

</select><br>

<label for="text-weight">Жирность текста:</label>

<select id="text-weight">

<option value="normal">Обычный</option>

<option value="bold">Жирный</option>

</select><br>

<button id="apply-styles">Применить стили</button>

</div>

<script src="../node\_modules/@joint/plus/joint-plus.js"></script>

<script src="../js/index.js"></script>

<script src="../js/script.js"></script>

</body>

</html>

Листинг Б.2 – Подпрограмма index.js.

var namespace = joint.shapes;

var title = document.getElementById('title');

var diagram\_name="diagram.json";

title.textContent=diagram\_name;

var canvas = document.getElementById('canvas');

var link = new namespace.standard.Link({

labels: {

attrs: {

text: {

text: 'Hello, World!'

}

}

},

router: { name: 'manhattan' }, // выбор маршрутизатора для гибкости связи

connector: { name: 'normal' }, // выбор типа коннектора для стрелки

attrs: {

line: {

stroke: 'black', // цвет стрелки

strokeWidth: 1, // ширина стрелки

targetMarker: {

type: 'path',

d: 'M 10 -5 0 0 10 5 Z', // форма маркера на конце стрелки

},

},

},

})

var graph = new joint.dia.Graph({}, { cellNamespace: namespace });

var paper = new joint.dia.Paper({

el: canvas,

model: graph,

width: canvas.clientWidth,

height: canvas.clientHeight,

gridSize: 10,

drawGrid: true,

background: {

color: '#FFF'

},

cellViewNamespace: namespace,

defaultLink: link,

linkPinning: false,

});

var paperSmall = new joint.dia.Paper({

el: document.getElementById('radar'),

model: graph,

width: 200,

height: 200,

gridSize: 1,

interactive: false,

cellViewNamespace: namespace

});

paperSmall.scale(0.25);

var commandManager = new joint.dia.CommandManager({ graph });

var selection = new joint.ui.Selection({

paper: paper

});

var clipboard = new joint.ui.Clipboard({

paper: paper

});

var currentElement;

const snaplines = new joint.ui.Snaplines({ paper });

// событие нажатия на любое место в paper

paper.on('blank:pointerdown', selection.startSelecting);

function initializeKeyboardShortcuts() {

this.keyboard = new joint.ui.Keyboard();

this.keyboard.on({

'ctrl+c': () => {

// копирование элементов и их стрелок

clipboard.copyElements(this.selection.collection, this.graph);

},

'ctrl+v': () => {

const pastedCells = this.clipboard.pasteCells(this.graph);

const elements = pastedCells.filter((cell) => cell.isElement());

selection.collection.reset(elements);

},

'ctrl+x shift+delete': () => {

clipboard.cutElements(this.selection.collection, this.graph);

},

'delete backspace': (evt) => {

graph.removeCells(this.selection.collection.toArray());

},

'ctrl+z': () => {

commandManager.undo();

selection.collection.reset([]);

},

'ctrl+y': () => {

commandManager.redo();

selection.collection.reset([]);

},

});

}

function createElement(shape) {

…

}

function createToolsForElement(element){

joint.elementTools.InfoButton = joint.elementTools.Button.extend({

name: 'info-button',

options: {

markup: [{

tagName: 'circle',

selector: 'button',

attributes: {

'r': 7,

'fill': '#001DFF',

'cursor': 'pointer'

}

}, {

tagName: 'path',

selector: 'icon',

attributes: {

'd': 'M -2 4 2 4 M 0 3 0 0 M -2 -1 1 -1 M -1 -4 1 -4',

'fill': 'none',

'stroke': '#FFFFFF',

'stroke-width': 2,

'pointer-events': 'none'

}

}],

x: '100%',

y: '0%',

offset: {

x: 0,

y: 0

},

rotate: true,

action: function(evt) {

alert('View id: ' + this.id + '\n' + 'Model id: ' + this.model.id);

}

}

});

var boundaryTool = new joint.elementTools.Boundary({

focusOpacity: 0.9,

rotate: true,

padding: 10,

});

var removeButton = new joint.elementTools.Remove();

var infoButton = new joint.elementTools.InfoButton();

var toolsView = new joint.dia.ToolsView({

tools: [

boundaryTool,

infoButton,

]

});

var elementView = element.findView(paper);

elementView.addTools(toolsView);

paper.on('element:mouseenter', function(elementView) {

elementView.showTools();

const halo = new joint.ui.Halo({

cellView: elementView ,

});

halo.render();

});

paper.on('element:mouseleave', function(elementView) {

elementView.hideTools();

});

paper.on('element:pointerclick', function(elementView) {

const freeTransform = new joint.ui.FreeTransform({ cellView: elementView });

freeTransform.render();

});

var verticesTool = new joint.linkTools.Vertices();

var segmentsTool = new joint.linkTools.Segments();

var sourceAnchorTool = new joint.linkTools.SourceAnchor();

var targetAnchorTool = new joint.linkTools.TargetAnchor();

var linkTools = new joint.dia.ToolsView({

tools: [

verticesTool, segmentsTool,

sourceAnchorTool, targetAnchorTool,

boundaryTool, removeButton

]

});

graph.on('add', (cell) => {

if (cell.isElement()) return;

cell.findView(paper).addTools(linkTools);

});

var mask = joint.highlighters.mask;

// Обработчик события клика по элементу

paper.on('element:pointerclick', (elementView) => {

// Добавляем подсветку элементу с классом 'my-element-highlight'

mask.add(elementView, { selector: 'root' }, 'my-element-highlight', {

deep: true,

attrs: {

'stroke': '#FF4365', // Цвет обводки

'stroke-width': 3 // Толщина обводки

}

});

});

// Обработчик события клика по пустому месту

paper.on('blank:pointerclick', function() {

// Удаляем все подсветки со всех ячеек

graph.getCells().forEach(function(cell) {

mask.remove(cell.findView(paper));

});

elementView.hideTools();

});

// Обработчик события двойного клика по ссылке

paper.on('link:pointerdblclick', function(linkView, evt) {

// Редактирование метки ссылки

var index = Number(linkView.findAttribute('label-idx', evt.target));

joint.ui.TextEditor.edit(evt.target, {

cellView: linkView,

textProperty: ['labels', index, 'attrs', 'text', 'text'],

annotationsProperty: ['labels', index, 'attrs', 'text', 'annotations']

});

});

// Функция авторазмера элемента

function autoSize(element) {

var view = paper.findViewByModel(element);

var textVel = view.vel.findOne('text');

// Используем bounding box без трансформаций для корректного авторазмера

var bbox = textVel.getBBox();

// 16 = 2\*8 - это перевод, заданный через ref-x и ref-y для нашего элемента

element.resize(bbox.width + 16, bbox.height + 16);

}

// Слушатель изменений атрибутов ячейки для авторазмера

graph.on('change:attrs', function(cell) { autoSize(cell) });

// Добавление обработчика события двойного клика по элементу

paper.on('element:pointerdblclick', function(elementView, evt) {

joint.ui.TextEditor.edit(evt.target, {

cellView: elementView,

textProperty: ['attrs', 'label', 'text'],

annotationsProperty: ['attrs', 'label', 'annotations']

});

// Добавляем обработчик события keydown

document.addEventListener('keydown', function(event) {

// Проверяем, нажата ли клавиша "Enter" (код клавиши 13)

if (event.keyCode === 13) {

// Завершаем редактирование текста

joint.ui.TextEditor.close();

}

});

});

// Добавление обработчика события двойного клика по ссылке

paper.on('link:pointerdblclick', function(linkView, evt) {

// Редактирование метки ссылки

var index = Number(linkView.findAttribute('label-idx', evt.target));

joint.ui.TextEditor.edit(evt.target, {

cellView: linkView,

textProperty: ['labels', index, 'attrs', 'text', 'text'],

annotationsProperty: ['labels', index, 'attrs', 'text', 'annotations']

});

// Добавляем обработчик события keydown

document.addEventListener('keydown', function(event) {

// Проверяем, нажата ли клавиша "Enter" (код клавиши 13)

if (event.keyCode === 13) {

// Завершаем редактирование текста

joint.ui.TextEditor.close();

}

});

});

}

// Экспорт SVG из бумаги

function exportSVG() {

paper.hideTools();

joint.format.toSVG(

paper,

(svg) => {

joint.util.downloadDataUri(

`data:image/svg+xml,${encodeURIComponent(svg)}`,

'joint-plus.svg'

);

},

{ useComputedStyles: false }

);

}

// Экспорт PNG из бумаги

function exportPNG() {

paper.hideTools();

joint.format.toPNG(paper, (dataURL, error) => {

if (error) {

console.error(error.message);

}

const lightbox = new joint.ui.Lightbox({

image: dataURL,

downloadable: true,

fileName: 'diagram.png'

});

lightbox.open();

paper.showTools();

}, {

padding: 10,

useComputedStyles: false,

grid: true

});

}

// Экспорт JPG из бумаги

function exportJPEG() {

paper.hideTools();

joint.format.toJPEG(paper, (dataURL, error) => {

if (error) {

console.error(error.message);

}

const lightbox = new joint.ui.Lightbox({

image: dataURL,

downloadable: true,

fileName: 'diagram.jpeg'

});

lightbox.open();

paper.showTools();

}, {

padding: 10,

useComputedStyles: false,

grid: true

});

}

// Функция для загрузки JSON файла

function loadJSONFromFile(callback) {

// Создаем input элемент для выбора файла

const input = document.createElement('input');

input.type = 'file';

input.accept = 'application/json';

// Добавляем обработчик события выбора файла

input.addEventListener('change', function() {

// Проверяем, что файл был выбран

if (input.files.length > 0) {

const file = input.files[0];

const reader = new FileReader();

// Обработчик события загрузки файла

reader.onload = function(event) {

// Парсим JSON из содержимого файла

const jsonData = JSON.parse(event.target.result);

// Вызываем колбэк с загруженными данными

callback(jsonData);

};

// Читаем содержимое файла как текст

reader.readAsText(file);

}

});

// Кликаем на input элемент, чтобы вызвать окно выбора файла

input.click();

}

function saveJSONToFile(jsonData, fileName) {

const jsonString = JSON.stringify(jsonData, null, 2);

var file = new Blob([jsonString], {type: 'application/json'});

var a = document.createElement("a"),

url = URL.createObjectURL(file);

a.href = url;

a.download = fileName;

document.body.appendChild(a);

a.click();

setTimeout(function() {

document.body.removeChild(a);

window.URL.revokeObjectURL(url);

}, 0);

}

function initializeToolbar(){

// Получаем все ссылки в боковой панели

const undoButton = document.getElementById('undo-button');

const redoButton = document.getElementById('redo-button');

const clearButton = document.getElementById('clear-button');

const deleteButton = document.getElementById('delete-button');

const saveSelect = document.getElementById('saveSelect');

const saveButton = document.getElementById('save-button');

const openButton = document.getElementById('open-button');

const textBoxButton = document.getElementById('textBox');

// Получаем все элементы с классом "geDropdownItem"

const dropdownItems = document.querySelectorAll(".geDropdownItem");

// Получаем элемент input range

var rangeInput = document.getElementById('line-width-input');

var rangeValue = document.getElementById('range-value');

var colorInput = document.getElementById('color-pick-input');

// Перебираем все найденные элементы и назначаем обработчик события на каждый из них

dropdownItems.forEach(item => {

item.addEventListener("click", function() {

const itemId = this.id; // Получаем id элемента, на который был клик

// В зависимости от id элемента выполняем соответствующее действие

switch(itemId) {

case 'open-option':

loadJSONFromFile(function(jsonData) {

// Подгружаем JSON данные в граф

graph.fromJSON(jsonData);

});

break;

case 'save-option':

saveJSONToFile(graph.toJSON(), 'diagram.json');

break;

case 'exit-option':

const dropdown = document.getElementById('file');

dropdown.classlist.toggle("hidden");

break;

case 'undo':

commandManager.undo();

break;

case 'redo':

commandManager.redo();

break;

case 'cut':

clipboard.cutElements(selection.collection, graph);

break;

case 'copy':

clipboard.copyElements(selection.collection, graph);

break;

case 'paste':

const pastedCells = clipboard.pasteCells(graph);

const elements = pastedCells.filter((cell) => cell.isElement());

// Make sure pasted elements get selected immediately. This makes the UX better as

// the user can immediately manipulate the pasted elements.

selection.collection.reset(elements);

break;

case 'delete':

graph.removeCells(selection.collection.toArray());

break;

case 'zoom':

// Получаем текущий масштаб бумаги

const currentScale = paper.scale().sx;

// Уменьшаем масштаб на 25%, но не меньше 25%

const newScale = Math.max(currentScale - 0.25, 0.25);

// Применяем новый масштаб к бумаге

paper.scale(newScale);

break;

case 'zoom out':

// Получаем текущий масштаб бумаги

const currentScale1 = paper.scale().sx;

// Увеличиваем масштаб на 25%, но не больше 175%

const newScale1 = Math.min(currentScale1 + 0.25, 1.75);

// Применяем новый масштаб к бумаге

paper.scale(newScale1);

break;

default:

break;

}

});

});

//Масштабирование страницы

// Получаем ссылки на кнопку и выпадающий список

const zoomButton = document.getElementById('zoom-scale');

const scaleDropdown = document.getElementById('scale-dropdown');

// Обработчик клика по кнопке масштабирования

zoomButton.addEventListener('click', function() {

// Переключаем видимость выпадающего списка

scaleDropdown.classList.toggle('open');

});

// Обработчик клика по варианту масштаба

scaleDropdown.addEventListener('click', function(e) {

// Проверяем, был ли клик по элементу с классом "scale-option"

if (e.target.classList.contains('scale-option')) {

// Получаем выбранный масштаб

const scale = parseFloat(e.target.dataset.scale);

// Применяем выбранный масштаб к странице

canvas.style.transform = `scale(${scale})`;

// Скрываем выпадающий список

scaleDropdown.classList.add('hidden');

}

});

// Обработчик для кнопки увеличения

document.getElementById('zoom-button').addEventListener('click', function() {

// Получаем текущий масштаб бумаги

const currentScale = paper.scale().sx;

// Уменьшаем масштаб на 25%, но не меньше 25%

const newScale = Math.max(currentScale - 0.25, 0.25);

// Применяем новый масштаб к бумаге

paper.scale(newScale);

});

// Обработчик для кнопки уменьшения

document.getElementById('unzoom-button').addEventListener('click', function() {

// Получаем текущий масштаб бумаги

const currentScale = paper.scale().sx;

// Увеличиваем масштаб на 25%, но не больше 175%

const newScale = Math.min(currentScale + 0.25, 1.75);

// Применяем новый масштаб к бумаге

paper.scale(newScale);

});

undoButton.addEventListener("click", function() {

commandManager.undo();

});

redoButton.addEventListener("click", function() {

commandManager.redo();

});

deleteButton.addEventListener("click", function() {

graph.removeCells(selection.collection.toArray());

})

textBoxButton.addEventListener("click", function() {

// Здесь можно выполнить нужные действия при клике на ссылку

// Например, можно получить значение атрибута data-shape

var textBlock = new joint.shapes.standard.TextBlock();

textBlock.resize(100, 100);

textBlock.position(100, 130);

textBlock.attr('root/title', 'joint.shapes.standard.TextBlock');

textBlock.attr('body/fill', 'lightgray');

textBlock.attr('label/text', 'Hyper Text Markup Language');

// Styling of the label via `style` presentation attribute (i.e. CSS).

textBlock.attr('label/style/color', 'red');

textBlock.addTo(graph);

});

clearButton.addEventListener("click", function() {

graph.clear();

});

saveButton.addEventListener("click", function() {

saveJSONToFile(graph.toJSON(), 'diagram.json');

});

openButton.addEventListener("click", function() {

loadJSONFromFile(function(jsonData) {

// Подгружаем JSON данные в граф

graph.fromJSON(jsonData);

});

});

saveSelect.addEventListener("change", function() {

const selectedOption = saveSelect.value;

// Определяем выбранную опцию и вызываем соответствующую функцию сохранения

switch(selectedOption) {

case 'json':

saveJSONToFile(graph.toJSON(), 'diagram.json');

break;

case 'svg':

exportSVG()

break;

case 'png':

exportPNG()

break;

case 'jpg':

exportJPEG();

break;

default:

break;

}

});

colorInput.value = '#ffffff'; // Устанавливаем значение в формате цвета CSS

rangeInput.value = 10;

rangeValue.textContent = rangeInput.value; // Показываем значение

// Обработчик изменения значения range

rangeInput.addEventListener("input", function() {

rangeValue.textContent = rangeInput.value; // Обновляем отображаемое значение

});

// Обработчик события input для элемента выбора цвета

colorInput.addEventListener("input", function() {

paper.updateBackgroundColor(colorInput.value);

});

// Обработчик события изменения значения input range

rangeInput.addEventListener('input', function() {

// Получаем новое значение из элемента range

var newSize = parseInt(this.value);

// Обновляем grid.size

paper.setGridSize(newSize)

// Обновляем отображение сетки

paper.update();

});

var lineConnectionTypeSelect = document.getElementById('line-connection-type');

lineConnectionTypeSelect.addEventListener('change', function() {

var selectedValue = lineConnectionTypeSelect.value;

// Установка типа соединения для бумаги

switch (selectedValue) {

case 'manhattan':

link.router('manhattan')

break;

case 'metro':

link.router('metro')

break;

case 'normal':

link.router('normal')

break;

case 'orthogonal':

link.router('orthogonal')

break;

case 'rightAngle':

link.router({ name: 'orthogonal', args: { startDirection: 'left', endDirection: 'right' } })

break;

default:

break;

}

});

}

function initializePropertiesPanel(){

const propertiesPanel = document.getElementById("properties-panel");

// Обработчик события клика левой кнопкой мыши на фигуре

paper.on('element:pointerclick', function(cellView) {

// Проверяем, что клик был выполнен левой кнопкой мыши

if (cellView && cellView.model && cellView.model.isElement()) {

// Открываем панель свойств или выполняем другие необходимые действия

propertiesPanel.classList.toggle("open");

}

});

var propStrokeColor = document.getElementById('stroke-color');

var propFillColor = document.getElementById('fill-color');

var propLineWidth = document.getElementById('line-width');

var propWidth = document.getElementById('width');

var propHeight = document.getElementById('height');

var propLineStyle = document.getElementById('line-style');

var propTextContent = document.getElementById('text-content');

var propTextFont = document.getElementById('text-font');

var propTextWeight = document.getElementById('text-weight');

var propTextSize =document.getElementById('text-size');

var propApplyStyles = document.getElementById('apply-styles');

paper.on('element:pointerclick', function(elementView) {

currentElement = elementView.model;

propStrokeColor.value = currentElement.attr('body/stroke');

propFillColor.value = currentElement.attr('body/fill');

propLineWidth.value = currentElement.attr('body/strokeWidth');

propWidth.value = currentElement.attr('body/width');

propHeight.value = currentElement.attr('body/height');

var strokeDashArray = currentElement.attr('body/strokeDashArray');

// Проверяем значение strokeDashArray и устанавливаем соответствующий вариант в панели свойств

if (strokeDashArray === null || strokeDashArray === undefined) {

propLineStyle.value = 'solid'; // Сплошная линия

} else if (strokeDashArray === '5,5') {

propLineStyle.value = 'dashed'; // Пунктирная линия

} else if (strokeDashArray === '1,3') {

propLineStyle.value = 'dotted'; // Точечная линия

}

propTextContent.value = currentElement.attr('label/text');

propTextFont.value = currentElement.attr('label/fontFamily'); // font-family вместо fontSize

propTextWeight.value = currentElement.attr('label/fontWeight');

propTextSize.value = currentElement.attr('label/fontSize'); // Добавляем значение размера текста

colorInput.value = paper.color;

})

var propStrokeColor = document.getElementById('stroke-color');

var propFillColor = document.getElementById('fill-color');

var propLineWidth = document.getElementById('line-width');

var propWidth = document.getElementById('width');

var propHeight = document.getElementById('height');

var propLineStyle = document.getElementById('line-style');

var propTextContent = document.getElementById('text-content');

var propTextFont = document.getElementById('text-font');

var propTextWeight = document.getElementById('text-weight');

var propTextSize =document.getElementById('text-size');

var propApplyStyles = document.getElementById('apply-styles');

propApplyStyles.addEventListener('click', function() {

if(currentElement){

currentElement.attr('body/stroke', propStrokeColor.value);

currentElement.attr('body/strokeWidth', propLineWidth.value);

currentElement.resize(propWidth.value,propHeight.value);

currentElement.attr('body/fill', propFillColor.value);

if (propLineStyle.value === 'solid') {

currentElement.attr('body/strokeDashArray', null);

} else if ( propLineStyle.value === 'dashed') {

currentElement.attr('body/strokeDashArray', '5,5');

} else if (propLineStyle.value === 'dotted') {

currentElement.attr('body/strokeDashArray', '1,3');

}

currentElement.attr('label/text', propTextContent.value);

currentElement.attr('label/fontFamily', propTextFont.value);

currentElement.attr('label/fontWeight', propTextWeight.value);

currentElement.attr('label/fontSize', propTextSize.value );

}

})

}

function initializeSidebar(){

const sidebarLinks = document.querySelectorAll("#sidebar .geItem");

// Проходимся по каждой ссылке и добавляем прослушиватель событий

sidebarLinks.forEach(sidebarLink => {

sidebarLink.addEventListener("click", function() {

// Здесь можно выполнить нужные действия при клике на ссылку

// Например, можно получить значение атрибута data-shape

const shape = this.getAttribute("data-shape");

createElement(shape);

console.log("Clicked on shape:", shape);

});

});

}

initializeKeyboardShortcuts();

initializeToolbar();

initializePropertiesPanel();

initializeSidebar(