



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

**Институт цифровых
интеллектуальных систем**

**Кафедра
компьютерных систем управления**

Алферов Михаил Дмитриевич

Выпускная квалификационная работа по направлению подготовки
15.03.04 «Автоматизация технологических процессов и производств»,
профиль «Автоматизация технологических процессов и производств (в машиностроении)»
на тему:
«Разработка инструментария для создания и анализа логических схем»

Регистрационный номер № _____

Зав. кафедрой,
д.т.н., профессор

(подпись)

Мartiнов Георги Мартинов

Руководитель,
старший
преподаватель

(подпись)

Мартемьянова Наталья Сергеевна

Обучающийся:
студент гр. АДБ-20-06

(подпись)

Алферов Михаил Дмитриевич

Москва 2023



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

**Институт цифровых
интеллектуальных систем**

**Кафедра
компьютерных систем управления**

«УТВЕРЖДАЮ»

Заведующий кафедрой

_____ **Мартинов Г. М.**

«__» _____ 2023 г.

ЗАДАНИЕ

на выпускную квалификационную работу
по направлению подготовки 15.03.04 «Автоматизация технологических процессов и
производств»,
профиль: «Автоматизация технологических процессов и производств (в
машиностроении)»

Алферов Михаил Дмитриевич

группа АДБ-20-06

Тема: «Разработка инструментария для создания и анализа логических схем»

Тема утверждена приказом от «__» _____ 202__ г. № _____

Срок сдачи законченной ВКР на кафедру «__» _____ 202__ г.

Целью данной выпускной квалификационной работы является повышение эффективности и упрощение работы при выполнении заданий на практических занятиях по дисциплинам «Теория дискретных систем управления», «Теория автоматического управления».

В рамках выполнения выпускной квалификационной работы необходимо решить следующие задачи:

1. Сравнительный анализ имеющихся программных средств для создания логических схем и формирование требований к разрабатываемому решению.
2. Разработка архитектурной модели и алгоритма функционирования инструментария для создания и анализа логических схем.
3. Программная реализация инструментария для создания и анализа логических схем.
4. Тестирование разработанного инструментария и демонстрация применения в учебных задачах.

Руководитель,
старший преподаватель _____

Мартемьянова Наталья Сергеевна

Студент гр. АДБ-20-06 _____

Алферов Михаил Дмитриевич

ГРАФИК
выполнения выпускной квалификационной работы

Мероприятия	Сроки выполнения	Отметка руководителя о выполнении
Сравнительный анализ имеющихся программных средств для создания логических схем и формирование требований к разрабатываемому решению	до «01» ноября 2023 г.	
Разработка архитектурной модели и алгоритма функционирования инструментария для создания и анализа логических схем	до «20» декабря 2023 г.	
Программная реализация инструментария для создания и анализа логических схем	до «03» марта 2024 г.	
Тестирование разработанного инструментария и демонстрация применения в учебных задачах	до «26» апреля 2024 г.	

График составлен «03» октября 2023 г.

С графиком ознакомлен _____

Алферов М.Д.

Руководитель _____

старший преподаватель,

Мартемьянова Н.С.

Оглавление

Введение.....	6
Глава 1. Сравнительный анализ имеющихся программных средств для создания логических схем и формирование требований к разрабатываемому решению.	7
1.1. Анализ существующих программных решений	7
1.2. Анализ учебных материалов.....	13
Глава 2. Разработка архитектурной модели и алгоритма функционирования инструментария для создания и анализа логических схем.	16
2.1. Структура приложения	16
2.2. Прототип	19
2.3. Пример	24
Список используемых источников:	27

Введение

В выпускной квалификационной работе рассматривается актуальный вопрос, связанный с построением и анализом логических схем.

В настоящее время мы сталкиваемся с необходимостью эффективного использования информационных технологий в различных областях, в том числе и в образовании.

В рамках данного проекта предлагается разработка программного продукта, предоставляющего пользователю возможность создания и анализа логических схем. Это особенно актуально в условиях, когда логические схемы являются неотъемлемой частью образовательного процесса в многих учебных заведениях. В том числе и в нашем университете. Решение этой задачи имеет прямое практическое применение в студенческих работах, где подобные схемы используются для обучения работы с логикой в целом и для решения отдельных аналитических задач.

Создание программы для построения и сохранения логических схем предоставляет возможность упрощения процесса обучения, а также повышения эффективности научных исследований в областях логики, таких как «Основы нечеткой логики», «Теория автоматического управления» и других связанных с ней дисциплин.

В мире, где технологии стремительно развиваются и оказывают влияние на наш повседневный мир, создание подобного инструмента, становится особенно важным. Простой и интуитивно понятный интерфейс программы, специально адаптированный для обучения и ориентированный на студентов, делает этот инструмент неотъемлемым элементом в оптимизации процесса обучения. Она открывает простой доступ к созданию и анализу логических схем.

Глава 1. Сравнительный анализ имеющихся программных средств для создания логических схем и формирование требований к разрабатываемому решению.

1.1. Анализ существующих программных решений

Проведение обширного анализа существующих программных продуктов в области анализа и создания логических схем. Этот этап позволит выявить преимущества и недостатки существующих решений, а также определить тенденции и потребности пользователей.

Semestr.online:

Semestr.online представляет собой веб-платформу с простым уровнем сложности, и она бесплатна в использовании. Её главные преимущества заключаются в удобстве для образовательных целей и возможности совместной работы в режиме реального времени. Однако, стоит учитывать, что у этого инструмента могут быть ограничения в функциональности для более сложных задач, и также ограничена интеграция с другими приложениями. Semestr.online поддерживает форматы данных Docx и png.

Matlab/Simulink:

Matlab/Simulink представляет собой высокоуровневый инструмент средней сложности, доступный для Windows, MacOS и Linux. Он обладает широким набором математических функций и предоставляет удобную графическую среду для моделирования и симуляции динамических систем. Однако, у этого инструмента высокие системные требования, и он использует специфичный синтаксис, что может потребовать времени для изучения. Simulink также имеет свой собственный формат для дальнейшей работы, но предоставляет возможность создания пользовательских функций и расширений [\[1\]](#).

Scilab/Xcos:

Scilab/Xcos - это инструмент с высоким уровнем сложности,

поддерживаемый на Windows, MacOS и Linux, и он бесплатен. Этот инструмент также обладает широким набором математических функций, и Xcos представляет собой графическую среду для моделирования и симуляции систем. Однако, следует отметить ограниченную поддержку со стороны общества и разработчиков. Scilab/Xcos использует собственный формат для дальнейшей работы, но также предоставляет возможность создания пользовательских функций и расширений, включая расширения для работы с машинным обучением [\[21\]](#).

GNU Octave:

GNU Octave - это инструмент с высоким уровнем сложности, поддерживаемый на Windows, MacOS и Linux, и он бесплатен. Он также предоставляет широкий набор математических функций, а Forge Signal является графической средой для моделирования и симуляции систем. Однако, аналогично Scilab/Xcos, GNU Octave имеет ограниченную поддержку со стороны общества и разработчиков. Инструмент использует собственный формат для дальнейшей работы, и, как и предыдущие, обеспечивает возможность создания пользовательских функций и расширений [\[31\]](#).

Vissim:

Vissim - инструмент с высоким уровнем сложности, предназначенный для Windows и стоимостью около 130000 рублей. Он предоставляет широкий набор математических функций и позволяет пользователям создавать и настраивать логические схемы и модели с учетом специфических требований и параметров. Разработчики Vissim предоставляют документацию и обучающие программы пользователям. В то время как инструмент требует времени для изучения и освоения, он предоставляет возможность создания пользовательских функций и расширений [\[41\]](#).

Таблица 1 Сравнение существующих решений

Утилиты	Платформа	Уровень сложности	Цена (руб)	Преимущества	Недостатки	Совместимость с форматами данных	Дополнительные функции
Semestr.online	Web	Простой	Бесплатно	<ul style="list-style-type: none"> - Удобство и доступность для образовательных целей. - Возможность совместной работы в реальном времени. 	<ul style="list-style-type: none"> - Вероятные ограничения в функциональности для более сложных задач. - Ограниченная интеграция с другими приложениями. 	Docx, png	Есть онлайн калькулятор и автоматические функции для упрощения схемы
Matlab/Simulink	Windows/MacOS /Linux	Выше среднего	~228000	<ul style="list-style-type: none"> - Широкий набор математических функций - Simulink представляет удобную графическую среду для моделирования и симуляции динамических систем - Доступен широкий выбор инструментов для 	<ul style="list-style-type: none"> - Высокие системные требования - Специфичный синтаксис приложения 	Собственный формат для дальнейшей работы.	- Возможность создания пользовательских функций и расширений

				визуализации и моделирования			
Scilab/Xcos	Windows/MacOS /Linux	Выше среднего	Бесплатно	-Широкий набор математических функций -Xcos является графической средой для моделирования и симуляции систем	-Ограниченная поддержка со стороны общества и разработчиков	Собственный формат для дальнейшей работы.	-Возможность создания пользовательских функций и расширений -Расширения для работы с машинным обучением
GNU Octave	Windows/MacOS /Linux	Высокая	Бесплатно	-Широкий набор математических функций -Forge Signal является графической средой для моделирования и симуляции систем	-Ограниченная поддержка со стороны общества и разработчиков	Собственный формат для дальнейшей работы.	-Возможность создания пользовательских функций и расширений
Vissim	Windows	Выше среднего	~130000	-Широкий набор математических функций -Позволяет пользователю создавать и настраивать	- Требуется время для изучения и освоения.	Собственный формат для дальнейшей работы.	-Возможность создания пользовательских функций и расширений

				логические схемы и модели с учетом специфический требований и параметров -Разработчики предоставляют документацию и обучающие программы пользователям			
--	--	--	--	--	--	--	--

Ссылаясь на все изученные данные о существующих приложениях, можно оценить какое приложение нужно пользователю - студенту. Создание интуитивного и легкого в использовании приложение, разработанное специально для задач образования. Оно предоставляет возможность создавать логические схемы и автоматически рассчитывать логические выражения, что значительно упрощает процесс моделирования и анализа систем.

Преимущества:

Простота и интуитивность: приложение ориентировано на простоту использования, что делает его более доступным для новичков или тех, кто не имеет глубоких знаний в области математики. Интуитивный интерфейс может упростить процесс создания логических схем.

Специализированный функционал: Приложение специализируется исключительно на создании и анализе логических схем, оно может предоставить более узкоспециализированные инструменты и функции, что может быть полезным для пользователей, занимающихся именно этой областью.

Образовательная направленность: Приложение ориентировано на студентов и образовательные цели, оно будет включать обучающие материалы, уроки и ресурсы, что может помочь в понимании логических операций и работы с ними.

Легкость в создании и сохранении схем: Приложение предоставляет удобные инструменты для создания и сохранения логических схем, это может быть выгодно для студентов и исследователей, часто использующих подобные схемы в своей работе.

Бесплатность: Приложение бесплатное, оно может привлечь большее количество пользователей, особенно студентов с ограниченным бюджетом.

1.2. Анализ учебных материалов

Для успешной реализации поставленных задач предусматривается использование учебных материалов, которые обеспечат необходимые знания и понимание для разработки программного продукта. Это включает в себя материалы по алгебре логики, теории логических схем.

В основе логических схем лежат базовые булевы операции, такие как И, ИЛИ, НЕ, которые могут быть комбинированы для создания более сложных логических функций. Эти операции представляются символами и соединяются линиями, представляющими поток данных [\[5\]](#).

Одним из основных преимуществ логических схем является их интуитивная понятность и наглядность. Они позволяют инженерам и проектировщикам легко представлять и анализировать логику работы цифровых систем. Логические схемы также являются основой для проектирования и создания интегральных схем (ИС), которые объединяют множество логических элементов на одном кристалле.

Типичные элементы логических схем включают операции И, ИЛИ, НЕ, а также их комбинации, такие как И-НЕ, ИЛИ-НЕ и другие. Каждый элемент выполняет определенную логическую операцию и может быть использован для построения сложных цифровых устройств.

Отрицание (INV, NOT)

Отрицание (инверсия) - унарная операция над суждениями, результатом которой является суждение, «противоположное» исходному. Синоним: логическое «НЕ», инверсия [\[6\]](#).

Таблица 2 Отрицание

A	INV(A)
0	1
1	0

Конъюнкция (AND)

Конъюнкция - логическая операция, по своему применению максимально приближённая к союзу «и». Синонимы: логическое «И», логическое умножение, или просто «И».

Таблица 3 Конъюнкция

A	B	AND(A, B)
0	0	0
0	1	0
1	0	0
1	1	1

Штрих Шеффера (NAND)

Штрих Шеффера- инверсия результата конъюнкции, возвращает истину, если хотя бы один из операндов ложен.

Таблица 4 Штрих Шеффера

A	B	NAND(A, B)
0	0	1
0	1	1
1	0	1
1	1	0

Дизъюнкция (OR)

Дизъюнкция - логическая операция, по своему применению максимально приближённая к союзу «или» в смысле «или то, или это, или оба сразу». Синоним: логическое «ИЛИ».

Таблица 5 Дизъюнкция

A	B	OR(A, B)
0	0	0
0	1	1
1	0	1
1	1	1

Стрелка Пирса (NOR)

Стрелка пирса - инверсия результата дизъюнкции, возвращает истину только если оба операнда ложны.

Таблица 6 Стрелка Пирса

A	B	NOR(A, B)
0	0	1
0	1	0
1	0	0
1	1	0

Исключающее ИЛИ (XOR)

Исключающее ИЛИ - возвращает истину, если количество истинных операндов нечетно.

Таблица 7 Исключающее ИЛИ

A	B	XOR(A, B)
0	0	0
0	1	1
1	0	1
1	1	0

Глава 2. Разработка архитектурной модели и алгоритма функционирования инструментария для создания и анализа логических схем.

2.1. Структура приложения

Чтобы разработать структуру приложения надо выбрать какой подход разработки выбрать. В случае написания данного приложения наилучшим вариантом будет использовать спиральную модель разработки.

В этом случае фазы «анализ требований – проектирование – реализация – тестирование» выполняется более одного раза. Предпосылки к итерациям: предупреждение рисков (которые можно предугадать и которые на начальном этапе нельзя было предвидеть) необходимость предоставить руководителю проекта частичную версию проекта для получения отзывов и пожеланий; необходимость выполнять промежуточные интеграции, не откладывая эту фазу на самый конец. Общая идея – на каждом витке строить очередную версию программы, используя в качестве ее основы предыдущую версию.

Достоинства: возможность сбора метрических характеристик – т.е. сколько нужно времени на разработку очередной версии.

Недостатки: сложность поддержки целостности документации – каждая версия программного кода должна быть документирована. Управление документацией еще больше усложняется, когда следующая фаза начинается до завершения предыдущей.

После выбора модели разработки выбираем паттерн проектирования для нашей программы. В моём случае это будет паттерн «Стратегия».

Стратегия, Strategy — поведенческий паттерн проектирования, предназначенный для определения семейства алгоритмов, инкапсуляции каждого из них и обеспечения их взаимозаменяемости. Это позволяет выбирать алгоритм путем определения соответствующего класса. Паттерн

Strategy позволяет менять выбранный алгоритм независимо от объектов-клиентов, которые его используют [\[7\]](#).

Применимость:

Имеется много родственных классов, отличающихся только поведением. Стратегия позволяет сконфигурировать класс, задав одно из возможных поведений;

Необходимо иметь несколько разных вариантов алгоритма. Например, можно определить два варианта алгоритма, один из которых требует больше времени, а другой - больше памяти. Стратегии разрешается применять, когда варианты алгоритмов реализованы в виде иерархии классов;

В алгоритме содержатся данные, о которых клиент не должен «знать». Используется паттерн стратегия, чтобы не раскрывать сложные, специфичные для алгоритма структуры данных [\[8\]](#);

В классе определено много поведений, что представлено разветвленными условными операторами. В этом случае проще перенести код из ветвей в отдельные классы стратегий.

После выбора паттерна проектирования можно делать UML схему структуры приложения.

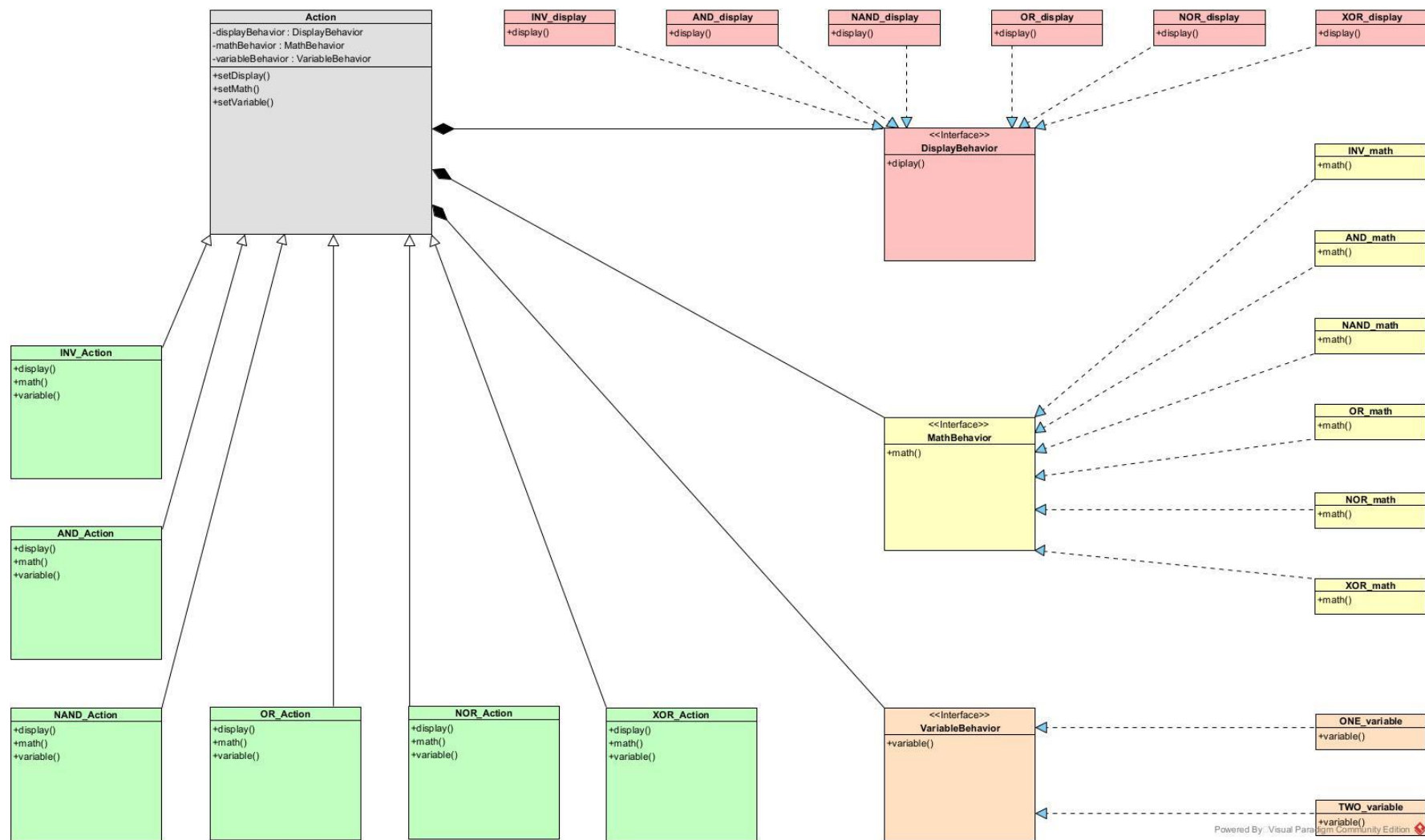


Рисунок 1 UML диаграмма работы кнопок действий

2.2. Прототип

Прототип играет важную роль в разработке программного продукта, предоставляя возможность предварительного визуального представления интерфейса и взаимодействия пользователя с приложением. Этот этап проектирования позволяет разработчикам и заказчикам оценить функциональность, эргономику и общее визуальное восприятие будущего приложения. В данном разделе мы рассмотрим процесс создания прототипа для нашего приложения по анализу и созданию логических схем.

Прототипирование предоставляет возможность визуализации основных элементов интерфейса, структуры приложения, а также взаимосвязей между различными компонентами. Это позволяет не только уточнить дизайн и функциональность, но также обнаружить потенциальные слабые места или улучшить пользовательский опыт до начала фактической разработки [\[9\]](#).

В данном разделе мы представим пример прототипа, охватывающего ключевые аспекты взаимодействия пользователя с приложением по анализу и созданию логических схем. Рассмотрим основные компоненты интерфейса, их визуальное оформление, а также способы взаимодействия пользователя с каждым элементом. Такой подход позволит нам детально проработать пользовательский опыт и обеспечить удобство использования приложения на каждом этапе его развития.

Общее строение приложения

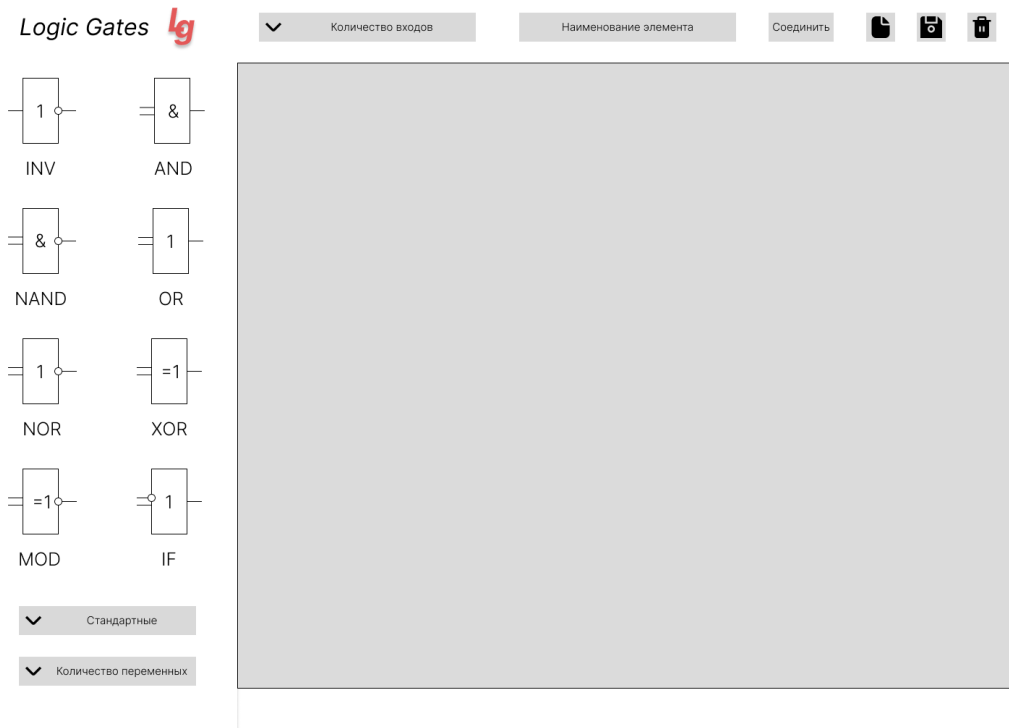


Рисунок 2 Визуальное представление

Рассмотрим построение приложение более подробно:

1. Кнопки элементов

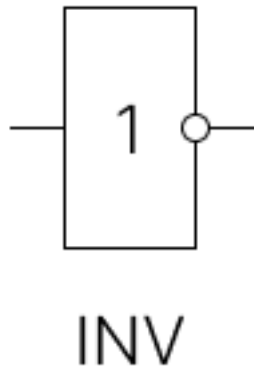


Рисунок 3 Кнопка элементов

Основное взаимодействие пользователя с нашим приложением будет реализовано через интуитивно понятные и легко доступные элементы управления. Ключевыми компонентами этого взаимодействия будут кнопки элементов и действий, которые пользователь может перетаскивать (drag and drop) на рабочее пространство для создания логических схем.

Функция drag and drop значительно улучшит пользовательский опыт, делая создание и анализ логических схем более удобным и интуитивным. Пользователи смогут легко выбирать логические элементы, операторы и другие действия, а затем перетаскивать их на рабочее пространство для пошагового построения схемы.

Такой подход позволит пользователям визуализировать логические операции, просматривать взаимосвязи между элементами и мгновенно видеть результаты своих действий. Это не только упростит процесс работы с приложением, но и сделает его более привлекательным для пользователей с различным уровнем опыта в алгебре логики.

2. Кнопки выбора

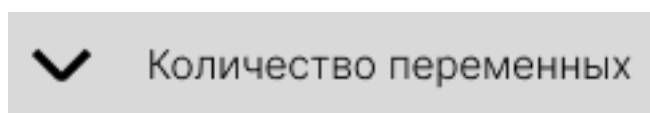


Рисунок 4 Кнопка выбора

Кнопки выбора в нашем приложении выполняют важную роль, предоставляя пользователю контроль и управление над процессом создания логических схем. Эти кнопки обеспечивают пользователю выбор из ограниченного, но всегда актуального числа элементов, что способствует удобству и простоте использования приложения.

В контексте выбора логических элементов, кнопки предоставляют доступ к различным операторам, включая логические вентили, операторы сравнения и другие. Это обеспечивает пользователю широкий набор инструментов для построения сложных и выразительных логических схем.

Кнопки также управляют параметрами элементов, такими как количество входов и выбор переменных для работы. Например, пользователь может определить, сколько переменных будет участвовать в логической операции, а также выбрать тип переменных (стандартные или продвинутые), что дает дополнительную гибкость и персонализацию в

процессе работы.

Такой подход к кнопкам выбора способствует более эффективному и управляемому процессу создания логических схем, делая приложение подходящим как для начинающих, так и для опытных пользователей в области алгебры логики.

3. Кнопки «Открыть», «Сохранить» и «Очистить всё»



Рисунок 5 Кнопка Открыть



Рисунок 6 Кнопка Сохранить



Рисунок 7 Кнопка Очистить все

Кнопки "Открыть", "Сохранить" и "Очистить всё" предоставляют стандартные, но важные функции для удобства работы с приложением. Эти элементы управления создают у пользователя четкий и понятный механизм взаимодействия с данными и обеспечивают дополнительные возможности для эффективного использования приложения.

4. Рабочая зона

Рабочая зона в приложении представляет собой центральное пространство, где пользователь может создавать, редактировать и визуализировать логические схемы. Эта зона является ключевым элементом взаимодействия и обеспечивает удобное пространство для работы пользователя.

5. Кнопка «Наименование элемента»

Кнопка "Наименование элемента" предоставляет пользователю возможность присвоения имен каждому элементу логической схемы. Эта функциональность призвана улучшить понимание структуры схемы и обеспечить пользователю средство для более детальной идентификации элементов. Вот несколько основных аспектов и преимуществ этой кнопки: Индивидуальная идентификация, понимание логики, улучшенная читаемость, удобство работы.

6. Кнопка «Соединить»

Кнопка "Соединить" предоставляет пользователю инструмент для определения связей между переменными и действиями внутри логической схемы.

7. Поле для расчётов

Поле для расчета является элементом визуализации приложения. Его главная цель - отображение конечного результата, происходящего в результате составленной логической схемы.

Поле для расчета предоставляет пользователю удобное место для визуализации каждого этапа расчета логической схемы. Пользователь может следить за процессом шаг за шагом.

2.3. Пример

Примеры играют ключевую роль в понимании функционала и эффективности разрабатываемого программного продукта. В данном разделе представлен конкретный кейс использования проектируемого приложения для создания логических схем. Развертывание практического примера поможет визуализировать процесс работы с приложением, выявить его удобства и дать пользователям представление о том, как оно может быть интегрировано в их повседневную деятельность. Детальное рассмотрение примера проектирования логической схемы позволит лучше понять функционал приложения.

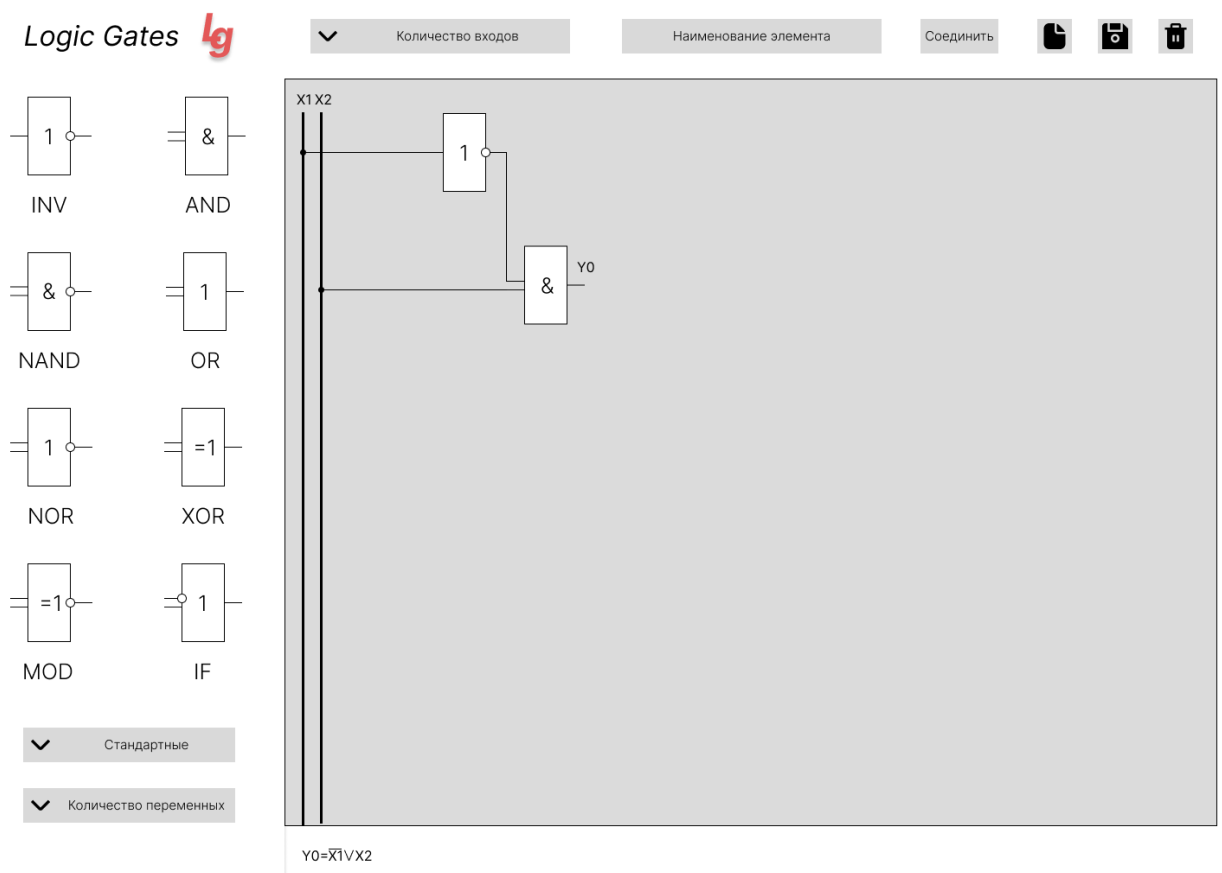


Рисунок 8 Пример работы

В данном случае рассматривается простой пример, который показывает работу с двумя переменными $x1$ и $x2$.

Действия:

1. Указываем количество переменных, в данном случае 2;
2. Нажимаем кнопку INV и соединяем с переменной x1;
3. Нажимаем кнопку AND и соединяем с INV и переменной x2.

После завершения процесса разработки логической схемы в приложении, получается готовая схема, которую можно сохранить в необходимом формате для последующего использования. Внизу интерфейса предоставляется выражение, произведенное на основе созданной схемы. Это выражение представляет собой логическое описание логики, заключенной в схеме, и может служить важным инструментом для более глубокого понимания работы разработанной логической конструкции.

Примером для моего проекта послужил одно из решений, которое было представлено выше, а именно Semestr.online. Но так как это web разработка, она не может предоставить такого же удобства использования как разрабатываемое приложение. Так же оно не имеет в себе инструментария для расчёта, это можно сделать только в отдельных приложениях или сервисах.

Количество переменных: 2 | Стандарт изображений элементов: IEC | Инверсные входы: отображать

1 INV & AND & NAND 1 OR 1 NOR =1 XOR =1 MOD 1 IF

Соединить | Свойства | Удалить | Действия ▾ | ?

Размеры графического полотна

Ширина	800	Высота	600
--------	-----	--------	-----

x1 x2 $\overline{x1}$ $\overline{x2}$

Рисунок 9 Работа web-сервиса Semest.online

Список используемых источников:

1. Дьяконов В., Строганова Е., Корнеев И., Бочкарева Е. Simulink 4. Специальный справочник. - 1-е изд. - СПб.: ПИТЕР, 2002. - 528 с.
2. Справка Scilab // [scilab.org](https://help.scilab.org/docs/5.5.2/ru_RU/index.html) URL: https://help.scilab.org/docs/5.5.2/ru_RU/index.html (дата обращения: 06.10.2023).
3. Алексеев Е.Р., Чеснокова О.В., Введение в Octave для инженеров и математиков. - 1-е изд. - М.: ALT Linux, 2012. - 368 с.
4. Дьяков В.П. VisSim+Mathcad+MATLAB. Визуальное математическое моделирование основы моделирования, описание библиотеки блоков, применение VisSim, пакеты расширения системы, 300 практических примеров. - 1-е изд. - М.: Солон-Пресс, 2004. - 386 с.
5. Никишечкин А.П. Теория дискретных систем управления. Учебное пособие. – М.: ИЦ ГОУ МГТУ «Станкин», 2006. – 242 с.
6. Никишечкин А.П. Дискретная математика и дискретные системы управления. 2-е изд., испр. и доп. – М.: Издательство Юрайт, 2018. – 298 с.
7. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно-ориентированного проектирования. - 2020. - 368 с.
8. Фримен Эр., Фримен Эл., Бейтс Б., Сьерра К Паттерны проектирования. - Питер, 2012. - 656 с.
9. Мандел, Т. Разработка пользовательского интерфейса [Электронный ресурс] / Т. Мандел; Пер. с англ. - Москва : ДМК Пресс, 2007. - 416 с., ил. - (Серия «Для программистов»). - ISBN 5-94074-069-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/407684> (дата обращения: 14.10.2023)