

Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Сибирский государственный университет телекоммуникаций и информатики»  
(СибГУТИ)

09.03.01 "Информатика и вычислительная техника"  
профиль "Программное обеспечение средств  
вычислительной техники и автоматизированных систем"

## ОТЧЕТ

По расчётно-графическому заданию  
по дисциплине «Визуальное программирование и человеко-машинное  
взаимодействие»

На кафедре Прикладной Математики и Кибернетики

Выполнил:

студент гр. ИП-116

Пожидаев И.О.

Проверил:

старший преподаватель

каф. ПМиК

\_\_\_\_\_/ Милешко А.В./

«\_\_» \_\_\_\_\_ 2023г.

Оценка \_\_\_\_\_

Новосибирск 2023 г

## Задание на РГР

Реализовать приложение-симулятор логических схем.

Работа состоит из следующих этапов:

1. Создание Use-Case диаграммы приложения. По окончании этапа должны быть построены Use-Case диаграммы.

2. Разработка графического интерфейса (схематичное изображение интерфейса и описание возможностей элементов, достижения сценариев описанных в Use-Case диаграмме посредством этих элементов). По окончании этапа должна быть построена схема интерфейса с подробным описанием элементов и достижения сценариев из use-case диаграммы.

3. Проектирование приложения - создание ER-диаграмм, диаграмм классов. По окончании этапа должны быть построены диаграммы классов с описанием (обязательно), ER-диаграммы (необязательно).

4. Разработка. При разработке используется TDD и упрощённый git flow (одна функциональность - одна ветка, коммиты в логических точках).

В репозитории приложения должен находиться отчёт по первым трём пунктам и проекты с исходным кодом и юнит-тестами.

*Главное окно* приложения состоит из холста, на котором размещаются логические элементы; дерева проекта, на котором отображается список схем проекта; панели элементов на котором располагаются логические элементы, которые можно добавлять на схему; меню управления из которого можно сохранить текущий проект, загрузить существующий проект, выйти из программы.

*Работа со схемой.* Для размещения элемента на схеме, нужно выбрать логический элемент на панели и кликнуть левой кнопкой мыши на холст в то место на которое нужно поместить логический элемент. При зажатой левой кнопкой мыши на логическом элементе, расположенном на холсте, его можно перетаскивать.

Для соединения логических элементов необходимо перетащить выход одного элемента на вход другого элемента или наоборот. При этом после соединения появляется линия между входом и выходом соединённых элементов. Для удаления соединения необходимо выбрать линию соединения левой кнопкой мыши и нажать клавишу delete. Для удаления элемента, необходимо выбрать удаляемый элемент левой кнопкой мыши и нажать клавишу delete.

Выходные сигналы логических элементов должны рассчитываться в реальном времени.

*Работа с проектом.* Проект имеет вид списка с верхним элементом - названием проекта, все остальные элементы названия схем. Схемы можно добавлять и удалять, но в проекте всегда должна быть минимум одна схема. Чтобы отредактировать схему нужно кликнуть на неё в списке два раза левой кнопкой мыши. Название проекта

можно отредактировать кликнув два раза левой кнопкой мыши на нём.

*Меню* должно включать четыре пункта: "Создать", "Открыть", "Сохранить", "выйти".

Пункт "Создать" - создаёт новый проект.

Пункт "Открыть" - открывает диалоговое окно открытия файла проекта, при выборе файла проекта, проект подгружается в приложение - его схемы отображаются в списке схем проекта, на холсте появляется отображение первой схемы в проекте.

Пункт "Сохранить" - открывает диалоговое окно сохранения проекта в файл. При выборе файла содержимое проекта сохраняется в него.

Пункт "Выйти" - закрывает приложение.

*Стартовое окно.* При старте приложения появляется окно в котором отображается список недавно открывавшихся проектов, отсортированных по дате открытия. Также есть кнопки "Создать новый проект", "Открыть проект", "Выйти", которые дублируют пункты меню "Создать", "Открыть", "Выйти" основного окна. При выборе проекта из списка стартового окна, он открывается в главном окне.

*Поддерживаемые логические элементы.* Приложение должно иметь поддержку логических элементов И, ИЛИ, НЕ, ИСКЛ-ИЛИ, а также элементов ВХОД и ВЫХОД. ВХОД позволяет по клику на нему левой кнопкой мыши поменять выходящее из него значение, ВЫХОД отображает значение сигнала, подающееся ему на вход.

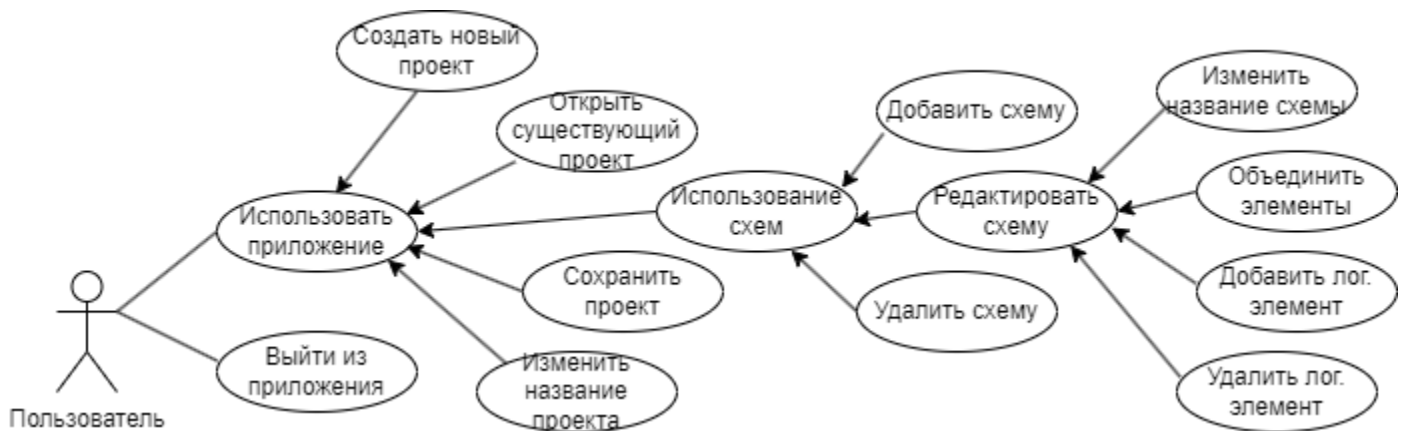
Вариант:

Формат хранения проекта – 2 – JSON

Формат хранения списка открывавшихся проектов – 1 – XML

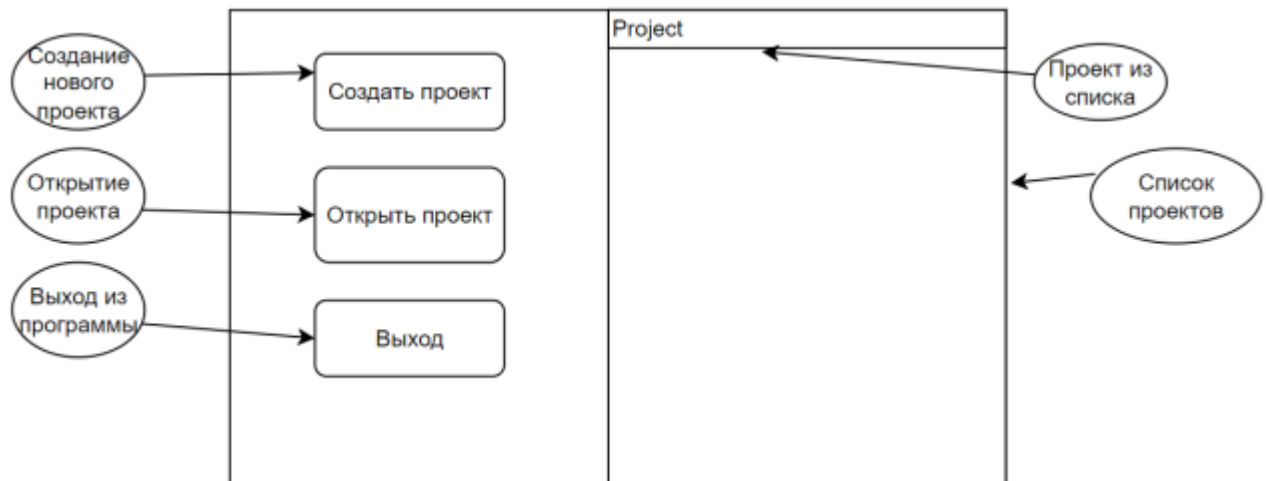
Дополнительный логический элементы – 2 – Демультимплексор

## Создание Use-Case диаграммы приложения



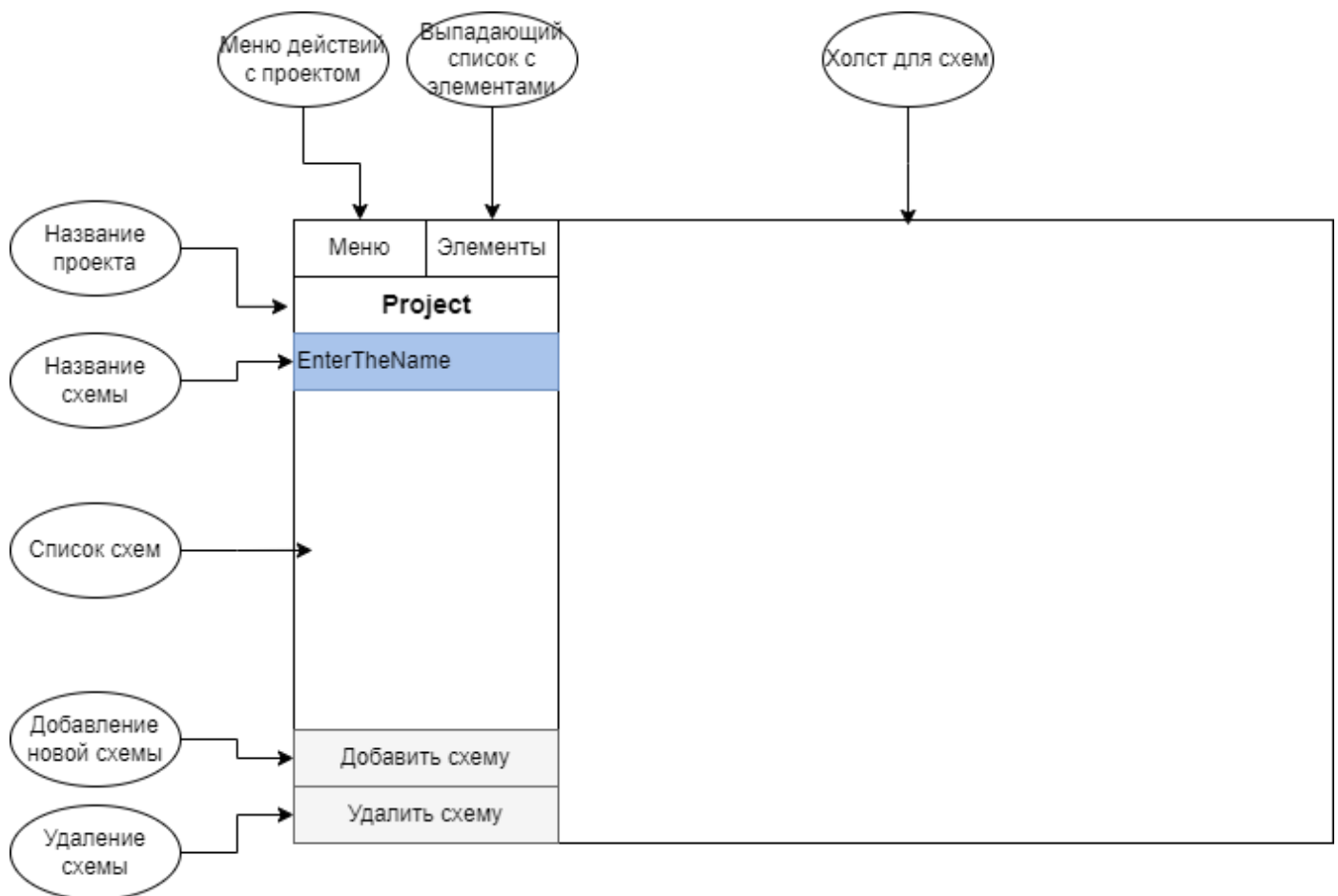
## Разработка графического интерфейса

### Начальное окно

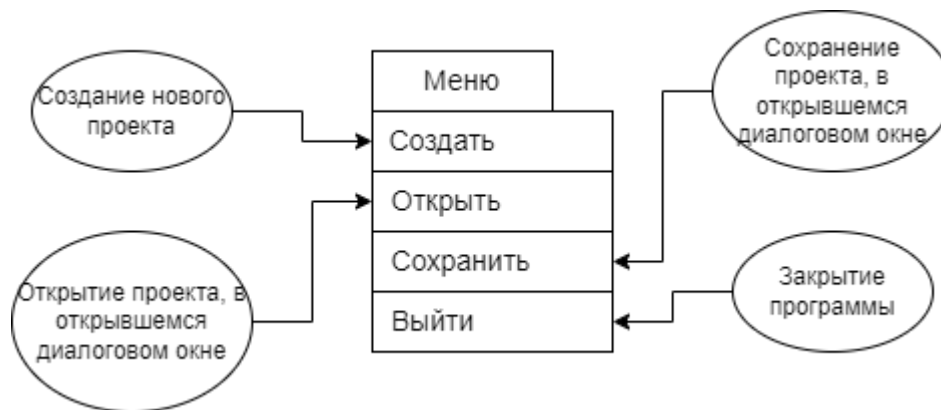


Начальное окно представлено 3-мя кнопками и списком проектов. Чтобы создать новый проект, следует нажать на кнопку “Создать новый проект” и вас перенесет в окно редактора схем. Чтобы открыть уже существующий проект, для начала нужно выбрать нужный проект из списка проектов и нажать на кнопку “Открыть проект”. И, наконец, чтобы выйти из приложения, пользователю необходимо нажать на кнопку “Выход”, после чего произойдет закрытие программы.

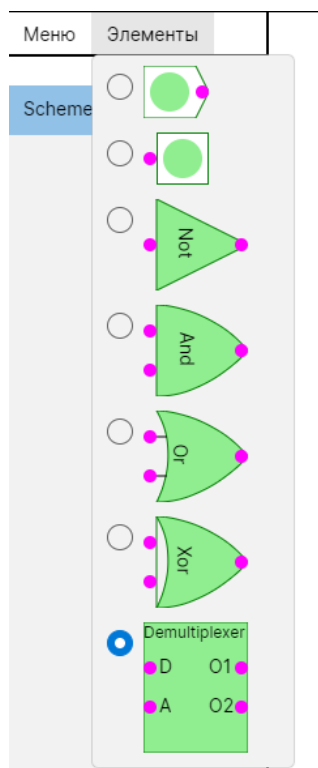
## Главное окно



Главное окно проекта состоит из холста и элемент управления. Элемент управления представлен следующими элементами: название схемы, которое можно менять двойным нажатием на имя; списком схем, между которыми можно переключаться одинарным нажатием; кнопкой “Добавить схему”, при нажатии на которую создается новая схема и она добавляется в список схем; кнопкой “Удалить схему”, при нажатии которой удаляется выбранная схема из списка проектов; меню, состоящее из меню действий с проектом и списком элементов. Рассмотрим “Меню”:



В “Меню” пользователь может создать новый проект, с помощью кнопки “Создать”, также используя кнопку “Открыть”, появляется диалоговое окно, в котором нужно выбрать нужный файл для открытия проекта. При нажатии на кнопку “Сохранить”, аналогичным образом открывается диалоговое окно, но теперь уже для сохранения файла. И кнопка “Выйти” просто закрывает программу. Перейдем к пункту “Элементы”



The diagram illustrates the architecture of a GUI framework, organized into several key packages and classes:

- LogicalElements**: Contains logic for various UI components.
  - NotGate**: Implements a NOT logic gate.
  - OutputElement**: Represents an output element with signal and connection properties.
  - InputElement**: Represents an input element with signal and connection properties.
  - Gate**: A base class for logic gates, including **XorGate**, **OrGate**, and **AndGate**.
  - TwoInputsGate**: A base class for gates with two inputs, including **AndGate** and **OrGate**.
- Element**: A base class for UI elements, including **Connector**, **Project**, and **ChangeStartPointEventArgs**.
- SerializableElements**: Contains serializable versions of the elements, such as **SerializableOutput**, **SerializableLogicalElement**, and **SerializableConnector**.
- ViewModels**: Contains the view models for the UI, including **MainUserControlViewModel**, **StartViewModel**, and **MainWindowViewModel**.
- Converters**: Contains classes for converting values, such as **FocusColorConverter** and **SignalToColorConverter**.
- Interfaces**: Includes **ValueConverter** and **ValueConverter**.

The diagram uses standard UML notation to show relationships: solid lines for associations, dashed lines for generalizations (Extends), and solid lines with open arrows for dependencies (Use). The diagram is a simplified version of the one in the image, focusing on the main classes and their relationships.