Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Поволжский государственный университет телекоммуникаций и информатики

Факультет №1

Кафедра «Программная инженерия» (ПрИ)

ДОПУСТИТЬ К ЗАЩИТЕ ЗАЩИЩЕН С ОЦЕНКОЙ\_\_\_\_\_\_\_\_

Руководитель \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ Руководитель \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ (подпись) (ФИО)` (подпись) (ФИО) \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_

(дата) (дата)

КУРСОВАЯ РАБОТА

по дисциплине Операционные системы и оболочки

наименование дисциплины (модуля)

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

наименование темы (при наличии)

ВЫПОЛНИЛ

студент ПрИ-21 Морзюков М.А.

(группа) (ФИО)

220392

(№ зачетной книжки)

Самара 2024

**РЕЦЕНЗИЯ**

на курсовую работу по дисциплине Операционные системы и оболочки

Студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Рецензент – руководитель КР:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О., степень, звание, должность)

*Добавить Drag and Drop. Переделать межпроцессное взаимодействие.*

*Определить критическую секцию. Переделать интерфейс.*

Результат рецензирования: допущен/не допущен к защите

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

(подпись) (ФИО руководителя)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

**Содержание**

[Задание 4](#_Toc167219076)

[Введение 5](#_Toc167219077)

[1 Проектирование приложения 6](#_Toc167219078)

[1.1 Разработка графического интерфейса 6](#_Toc167219079)

[1.2 Разработка программного обеспечения 11](#_Toc167219080)

[2 Разработка приложения 19](#_Toc167219081)

[2.1 Обеспечение приложения требуемым функционалом 19](#_Toc167219082)

[2.1.1 Настройка запуска программы 19](#_Toc167219083)

[2.1.2 Отображение файлов в таблице 20](#_Toc167219084)

[2.1.3 Конткестное меню 21](#_Toc167219085)

[2.1.4 Горячие клавиши 23](#_Toc167219086)

[2.1.5 Доработка таблицы с файлами 24](#_Toc167219087)

[2.1.6 Поисковая панель 25](#_Toc167219088)

[2.1.7 Панель меню 28](#_Toc167219089)

[2.1.8 Итоги базовой версии приложения 29](#_Toc167219090)

[2.1.9 Межпроцессное взаимодействие (задание 1) 30](#_Toc167219091)

[2.1.10 Отслеживание процессов (задание 2) 31](#_Toc167219092)

[2.1.11 Команды для терминала (задание 3) 31](#_Toc167219093)

[2.1.12 Дополнительные процессы (задание 4) 32](#_Toc167219094)

[2.1.13 Логирование 36](#_Toc167219095)

[2.1.14 Drag and Drop 36](#_Toc167219096)

[2.2 Тестирование 37](#_Toc167219097)

[3 Руководство пользователя 42](#_Toc167219098)

[3.1 Введение 42](#_Toc167219099)

[3.2 Назначение и условия применения 43](#_Toc167219100)

[3.3 Подготовка к работе. 43](#_Toc167219101)

[3.4 Описание операций. 44](#_Toc167219102)

[3.5 Аварийные ситуации. 46](#_Toc167219103)

[3.6 Рекомендации по освоению. 47](#_Toc167219104)

[Заключение 48](#_Toc167219105)

[Список использованных источников 49](#_Toc167219106)

[Приложение А 51](#_Toc167219107)

# Задание

Номер зачетной книжки – **220392**

Необходимо разработать программное обеспечение для работы с файлами и устройствами ввода/вывода, а также для мониторинга ресурсов. Задание курсовой работы состоит из следующих основных пунктов:

1. Разработка графического интерфейса.
2. Разработка программного обеспечения.
3. Обеспечение минимального функционала.
4. Добавление основного функционала.
5. Выбор стратегии тестирования и разработка тестов.
6. Формирование отчета по курсовой работе.

Задания для моего варианта:

1. Способ межпроцессного взаимодействия «Суперапп» и всплывающих окон: **Отображение файлов**
2. Отслеживание **всех процессов ОС, кроме процессов, запущенных из разрабатываемого приложения**.
3. В разрабатываемом приложение должен присутствовать терминал Linux и иметь следующий набор: **сетевые команды и команды для работы с файловой системой**.
4. Функционал для «Суперапп» распределяет преподаватель каждому отдельно:

* время работы ОС;
* ширину и высоту рамки окна приложения;
* обнаружение фактов переименования каталогов;

# Введение

На заре XXI века произошел значительный рост использования электронных устройств и цифровизации данных, охвативший все аспекты технологического прогресса, от программного обеспечения до аппаратных компонентов. Операционные системы стали фундаментальным звеном не только для ПК, но и для мобильных телефонов, планшетов и даже медицинской диагностической аппаратуры, обеспечивая критически важное взаимодействие между программами и аппаратной частью, а также удобство для пользователей. Понимание устройства и управления современными операционными системами является неотъемлемой частью подготовки профессионалов в области компьютерных наук и технологий.

XXI век часто упоминается как эра цифровой информации, где обработка и хранение данных достигли огромных масштабов, распространяясь на сотни тысяч серверов и суперкомпьютеров по всему миру.

**Актуальность** курсовой работы подтверждается скоростным развитием технологий и потребностью в глубоком понимании современных операционных систем, в особенности Linux. Это необходимо для обеспечения эффективного и безопасного взаимодействия между аппаратными и программными компонентами компьютерных систем. Кроме того, важной задачей является создание и тестирование новых приложений, способных оптимально использовать ресурсы системы и предоставлять пользователю необходимый функционал.

Основными **целями** курсовой работы являются укрепление и расширение теоретических знаний в области современных операционных систем, практическое применение усвоенных концепций и методов, а также освоение современных информационных технологий с целью их применения в профессиональной деятельности.

**Задачи** курсовой работы включают в себя освоение навыков проектирования и разработки компонентов операционных систем, организацию структуры программных средств и распределение функциональных задач между аппаратными и программными средствами. Также важным аспектом является приобретение практических навыков по написанию эффективного программного кода. Программа, разработанная студентом, должна соответствовать принятым стандартам описания программ.

**Объектом исследования** является операционная система Linux, а **предметом исследования** являются механизмы взаимодействия внутри этой операционной системы, а также разработка программного обеспечения для работы с файлами и устройствами ввода/вывода.

Работа организована таким образом, чтобы последовательно охватить все этапы проекта: от теоретической подготовки и проектирования до практической реализации и тестирования разработанного программного обеспечения. В заключении будут изложены основные выводы и результаты исследования, обобщающие основные аспекты выполненной работы.

# 1 Проектирование приложения

## 1.1 Разработка графического интерфейса

В данном разделе курсовой работы освещается процесс создания графического интерфейса пользователя с использованием веб-платформы Figma. Графический интерфейс выполняет важную функцию, делая взаимодействие пользователя с приложением интуитивно понятным и эффективным. Выбор Figma в качестве основного инструмента для проектирования интерфейса был сделан благодаря её продвинутым возможностям для разработки дизайна в режиме онлайн в реальном времени.

Процесс разработки интерфейса в Figma проходил в несколько этапов:

1. **Создание ручного наброска**: На первом этапе разработки были создан ручной набросок, который помог определить основные функциональные зоны и структуру навигации приложения (рис. 1.1). Этот этап способствовал формированию общего представления о внешнем виде интерфейса.

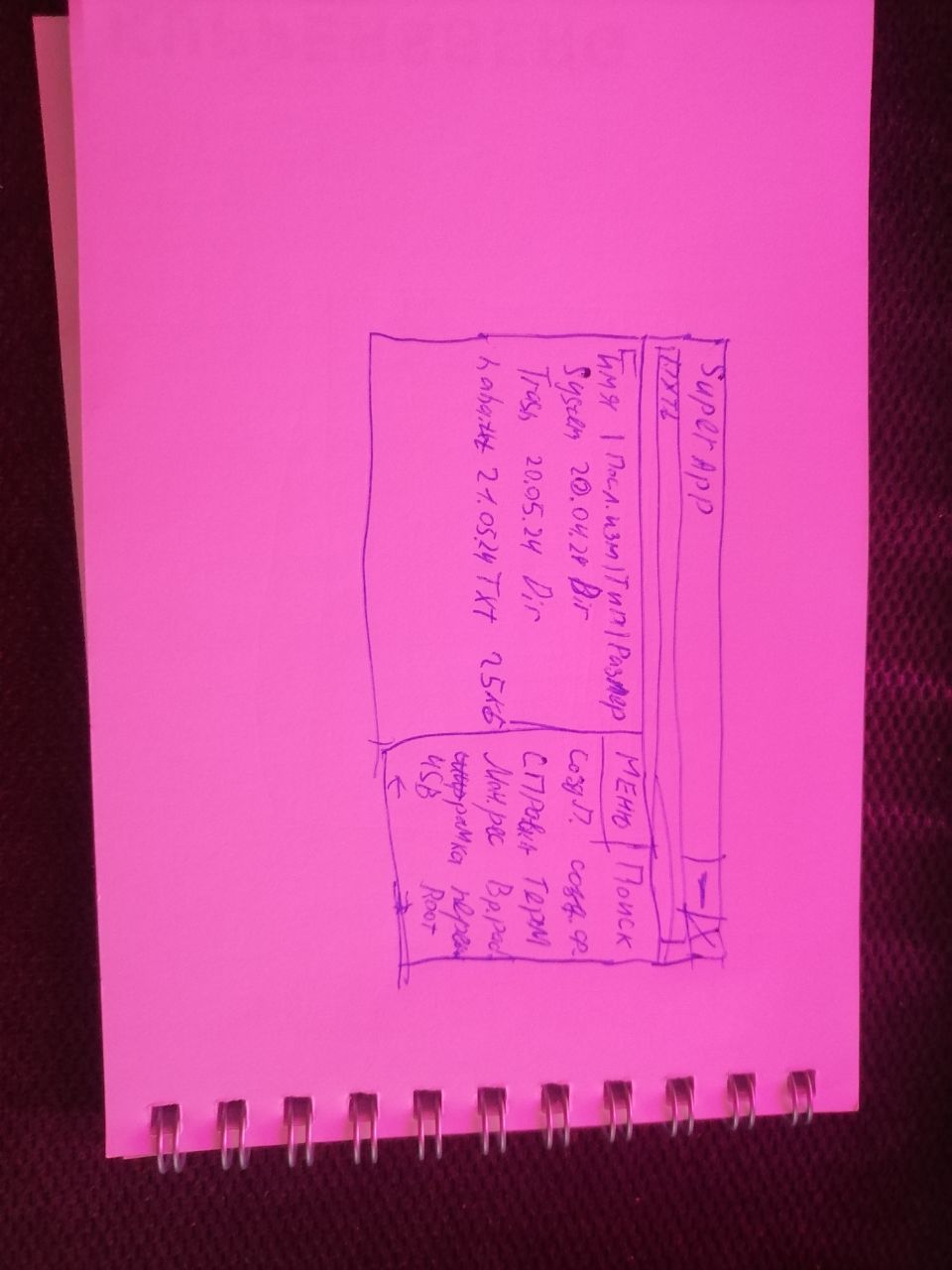


Рис. 1.1 – Ручной набросок

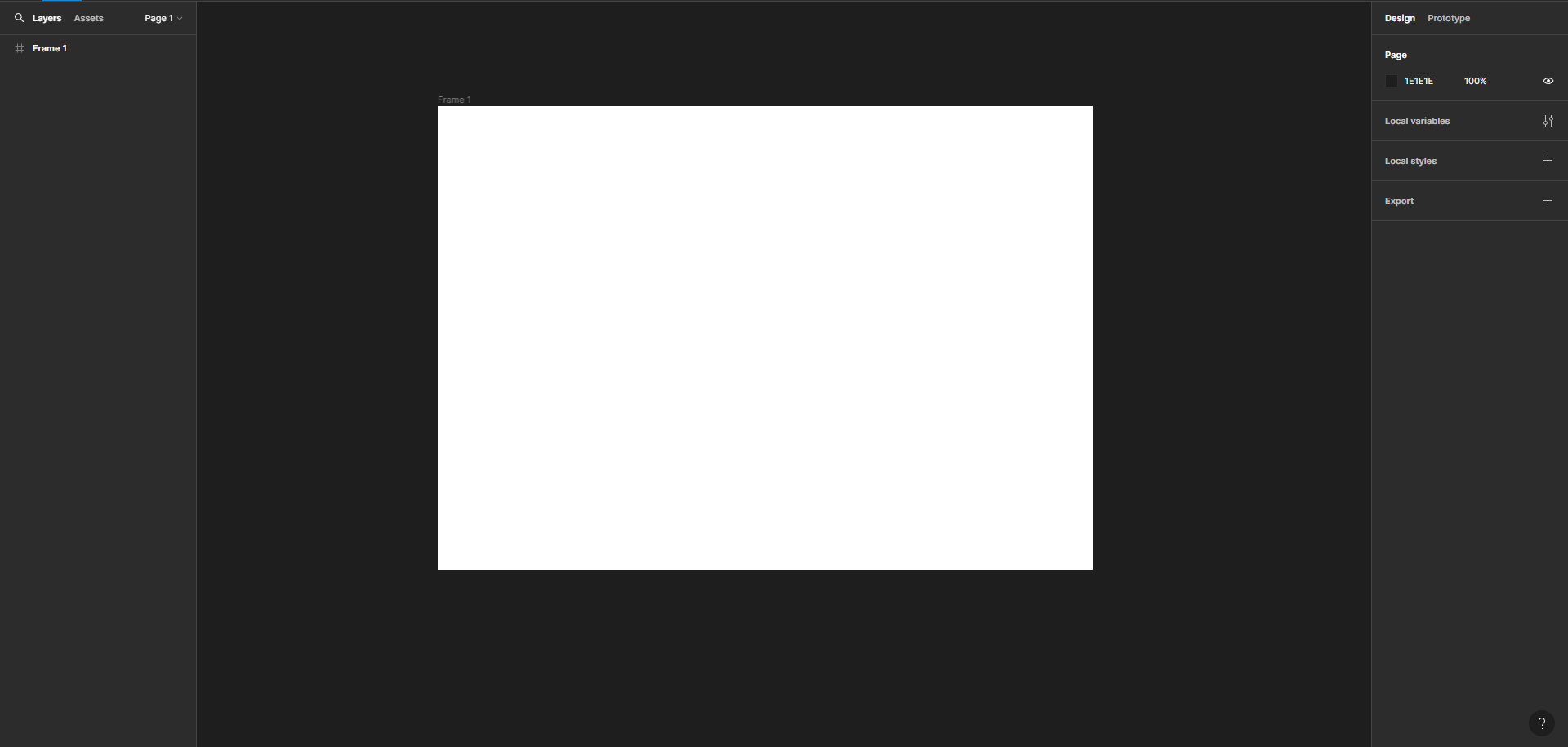
1. **Изучение возможностей Figma**: До начала работы над проектированием в Figma было проведено подробное изучение основных функциональностей и инструментов этой платформы. Это включало ознакомление с различными возможностями, инструментами рисования, создания компонентов и многими другими функциями. (Рис 1.2)

Рис. 1.2 – Интерфейс Figma

1. **Создание главного окна**: Используя скетчи как основу, было создано основное окно приложения в Figma (Рис. 1.3), где были определены основные элементы интерфейса. Процесс проектирования включал выбор цветовой палитры и стилей элементов.

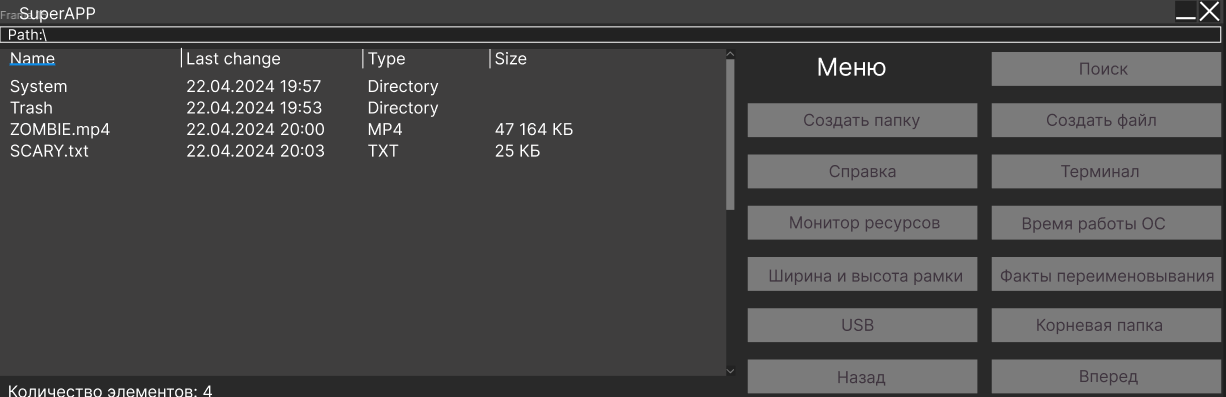


Рис. 1.3 – Главное окно для программы

1. **Описание дополнительных меню**: После завершения основного окна были разработано разработали контекстные команды (см. Рис. 1.4 – Рис. 1.5) и список горячих клавиш (см. Рис. 1.6).

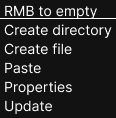


Рис. 1.4 – Контекстное меню при нажатии на пустое место в окне

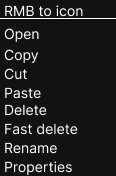


Рис. 1.5 – Контекстное меню при нажатии на файл / папку

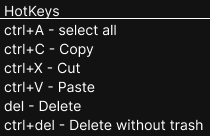


Рис. 1.6 – Горячие клавиши

1. **Создание окна терминала**: В Figma было создано окно терминала с текстовыми полями для ввода и вывода (Рис. 1.7).

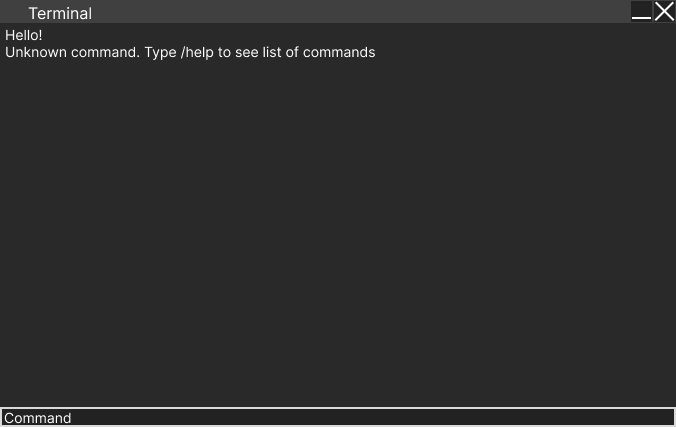


Рис. 1.7 – Окно терминала

Завершенный дизайн интерфейса, разработанный в Figma, отражает чистоту, функциональность и удобство для пользователя. Прикрепление скриншотов ключевых экранов интерфейса поможет наглядно представить достигнутые результаты и продемонстрировать эффективность выбранного дизайна.

Применение инструмента Figma действительно значительно упростило процесс создания графического интерфейса. Его возможности обеспечивают эффективное взаимодействие между дизайнерами и разработчиками, а также предоставляют возможность быстрого прототипирования. Этот проект прекрасно демонстрирует, как современные инструменты дизайна могут значительно ускорить процесс разработки и повысить качество конечного продукта.

## 1.2 Разработка программного обеспечения

Для создания приложения на Java с использованием библиотеки Swing была выбрана среда разработки IntelliJ IDEA, которая предлагает широкие возможности для удобной и эффективной работы над проектами. Этот выбор был сделан благодаря многочисленным функциям IDEA, способствующим продуктивности разработки.

В процессе разработки программы особое внимание было уделено созданию графического интерфейса пользователя, изначально спроектированного в Figma. Применение Java Swing позволило успешно адаптировать визуальные концепции дизайна к технической платформе приложения, сосредоточив внимание на визуальных аспектах, отложив интеграцию функциональных возможностей на более поздний срок.

Процесс адаптации дизайна из Figma в Java Swing начинался с детального рассмотрения всех дизайнерских элементов, включая расположение компонентов, цветовую схему, шрифты и другие пользовательские элементы управления. Благодаря обширным возможностям Java Swing для разработки пользовательских интерфейсов были предоставлены все необходимые инструменты для точного воспроизведения этих элементов. Основная задача этого этапа заключалась в достижении полного визуального соответствия с оригинальным макетом в Figma, что крайне важно для обеспечения последовательного пользовательского опыта.

При разработке графического интерфейса на Java Swing был применен модульный подход, в рамках которого были созданы отдельные классы для каждого компонента интерфейса (см. Рис. 1.8). Это решение сделало процесс разработки более простым, обеспечив удобство поддержки и возможность внесения изменений в будущем.

Каждый класс в архитектуре приложения был отведен под определенный элемент или функциональную область интерфейса. Такой подход позволил улучшить структуру кода, сделав его более понятным и легко управляемым.

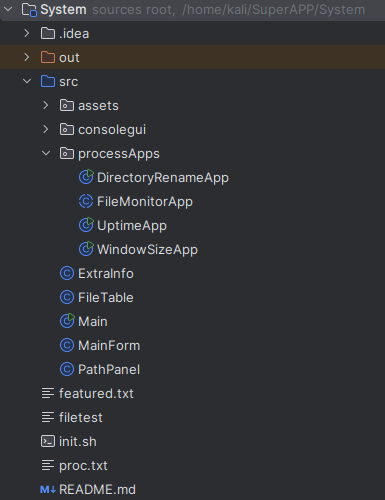


Рис 1.8 – Рабочая папка с классами для будущей программы

Каждый элемент интерфейса был представлен соответствующим виджетом Swing. Разработка началась с основного окна приложения (Рис. 1.9), затем перешла к созданию более мелких компонентов, таких как кнопки, поля ввода и другие интерактивные элементы. Одной из важных задач было сохранение всех пропорций и атрибутов стилей, чтобы обеспечить единообразный и профессиональный вид интерфейса.

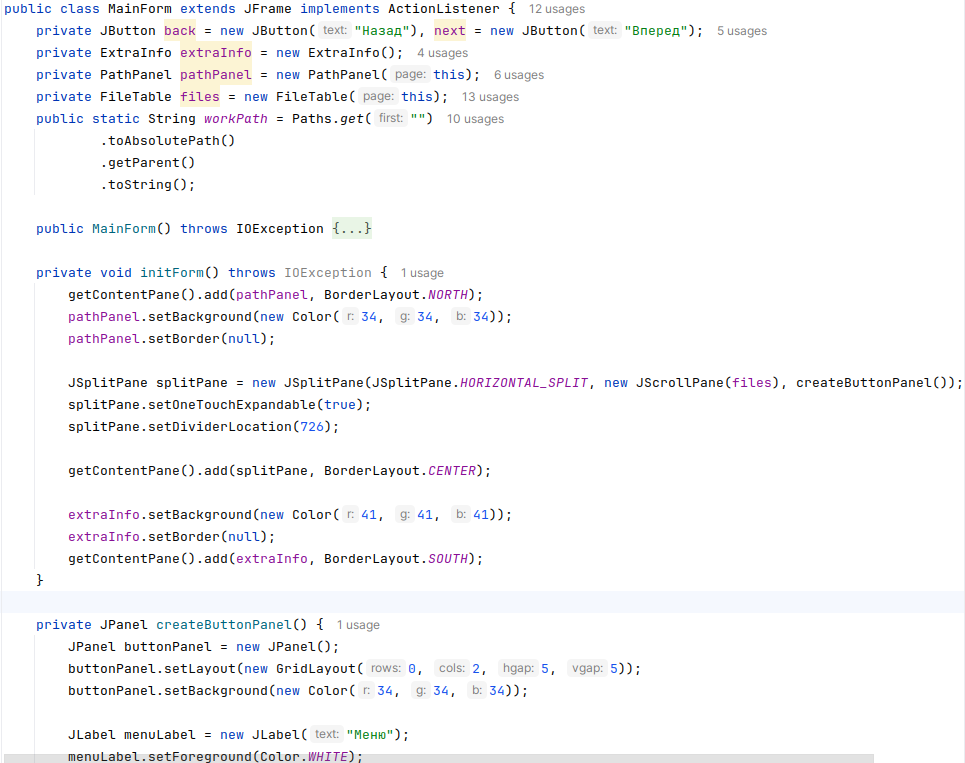


Рис 1.9 – Описание основного окна

Для панели меню был использован GridLayout, на который были добавлены кнопки (Рис 1.10).

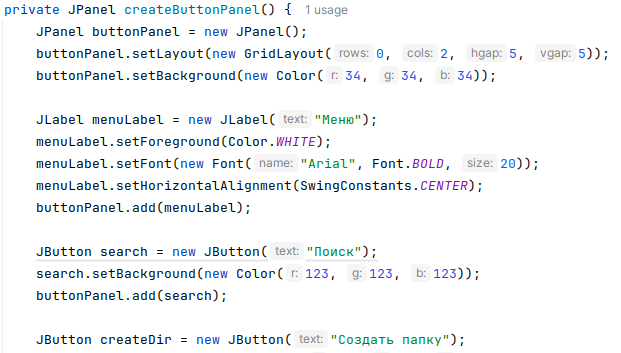


Рис 1.10 – Описание панели меню

В итоге мы получили нужную панель инструментов (Рис 1.11).

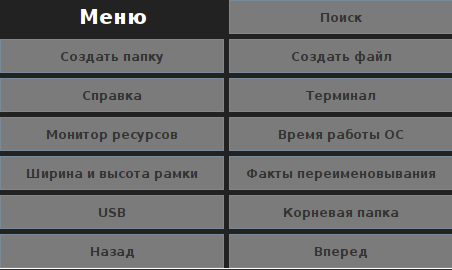


Рис 1.11 – Получившаяся панель меню

Для панели с текущим путем было использовано поле JTextField (Рис 1.12).



Рис 1.12 – Описание панели пути

Итоговая поисковая панель (Рис 1.13).



Рис 1.13 – Получившаяся поисковая панель

Для отображения файлов и директорий было принято решение использовать компонент JTable. В этой таблице будут отображаться файлы (Рис. 1.14 – Рис. 1.15) для обеспечения удобного и структурированного просмотра пользователем содержимого.

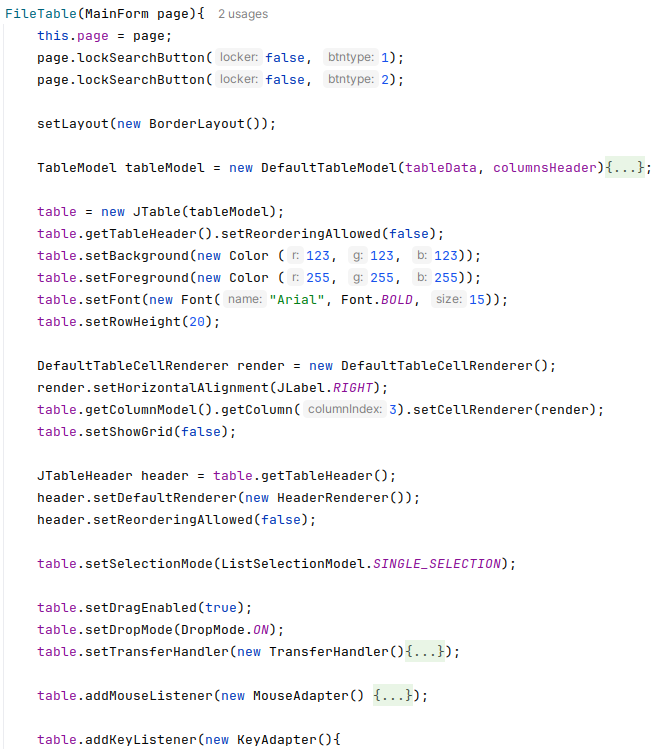


Рис 1.14 – Краткое описание кода окна с файлами

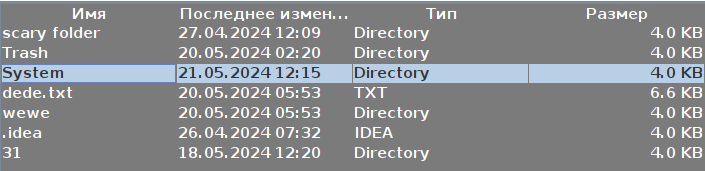


Рис 1.15 - Получившееся окно с файлами

Следом нужно было сделать нижнюю панель, на которой будет отображаться количество файлов в данной директории. Для этого был использован JLabel (Рис 1.16).

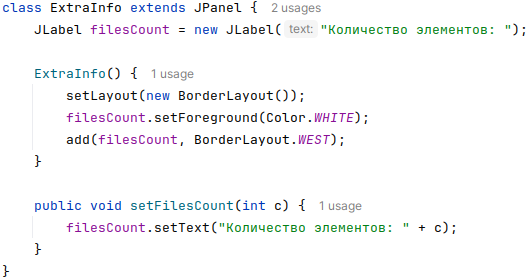


Рис 1.16 - Описание нижней панели.

Последняя панель была готова (Рис 1.17).



Рис 1.17 – Получившаяся нижняя панель

Благодаря использованию Java Swing удалось создать визуально привлекательный интерфейс, который почти полностью соответствует первоначальному дизайну, разработанному в Figma (см. Рис. 1.18). Этот процесс обеспечил гармоничное сочетание функциональности и эстетики, что повышает пользовательский опыт и удовлетворение от использования приложения.

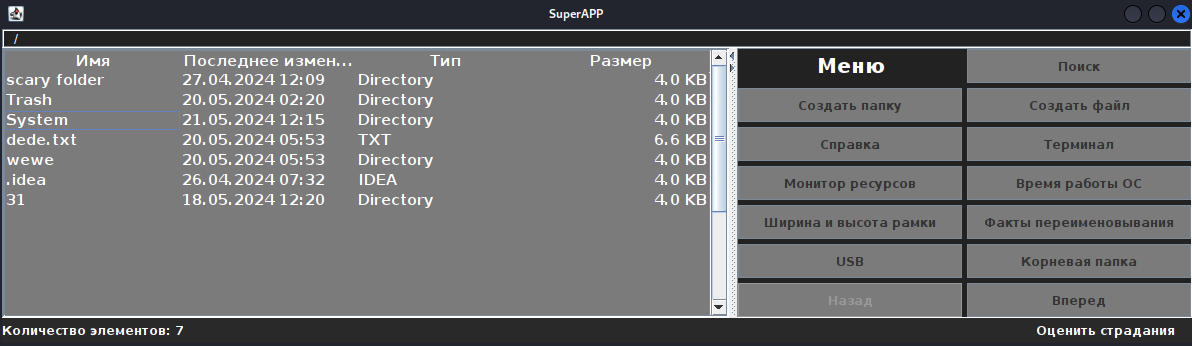


Рис 1.18– Итоговый интерфейс

Дальше нужно было добавить интерфейс для терминала, а так же контекстное меню.

Готовый терминал был уже создан на занятиях по лабораторным работам (Рис 1.19).

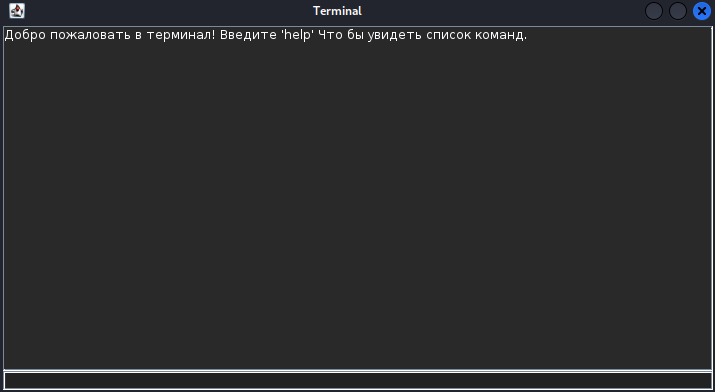


Рис 1.19 – Интерфейс терминала

Было принято решение добавить два контекстных меню. Первое меню активируется при нажатии на строку в таблице с файлами (см. Рис. 1.20). Второе меню появляется при нажатии на пустое место (см. Рис. 1.21). Для обоих контекстных меню был использован компонент JPopupMenu, в который входили элементы JMenuItem (см. Рис. 1.22, Рис. 1.23).

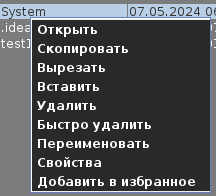


Рис 1.20 – Контекстное меню 1

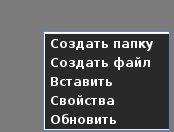


Рис 1.21 – Контекстное меню 2

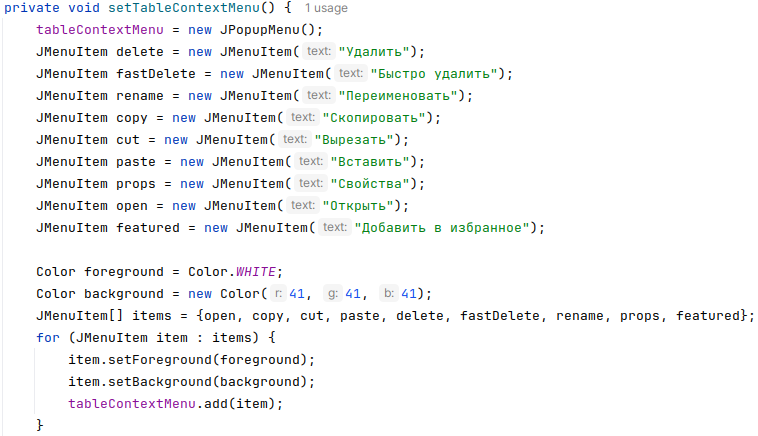


Рис 1.22 - Описание контекстного меню 1



Рис 1.23 – Описание контекстного меню 2

# 2 Разработка приложения

## 2.1 Обеспечение приложения требуемым функционалом

После завершения разработки графического интерфейса пользователя следующим шагом было внедрение функциональности в приложение. Одной из ключевых задач было создание и настройка исполняемого файла для системы Linux. Это решение позволило пользователям запускать приложение непосредственно через командную строку операционной системы, что упрощало процесс использования программы и делало его более нативным для пользователей Linux.

### 2.1.1 Настройка запуска программы

Для настройки среды перед запуском приложения был разработан специальный скрипт, включающий необходимые команды. Одним из важных этапов подготовки к запуску было установление прав доступа к файлам системы с помощью команды chmod a+rx. Эта команда предоставляет чтение и выполнение всех файлов в указанной директории всем пользователям, что повышает безопасность и предотвращает случайные изменения или удаления критически важных файлов.

Команда chmod a+rx была включена в скрипт, который выполняется перед компиляцией программы. Такой подход гарантирует, что все необходимые файлы будут защищены до начала работы приложения. Однако использование этой команды требует прав суперпользователя, поэтому предполагается, что скрипт будет запущен от имени пользователя с соответствующими привилегиями.



Рис 2.1 – Исполняемый файл для запуска приложения

### 2.1.2 Отображение файлов в таблице

Далее была реализована функциональность отображения файлов и папок в виде таблицы на форме FileList (Рис. 2.2). Таблица разделена на несколько столбцов: Name, Last change, Type и Size. Методы updateTable() и setDirFiles() используются для обновления таблицы, отображая в ней файлы из текущей директории. Для изменения текущей директории используются методы pathFieldSwitch(), dirSwitchForward(), dirSwitchBack(), dirSwitchToRoot() и dirSwitch(). Методы getFileTime(), makeFileSize() и getFileExtension() добавляют дополнительную информацию о файлах в таблицу.

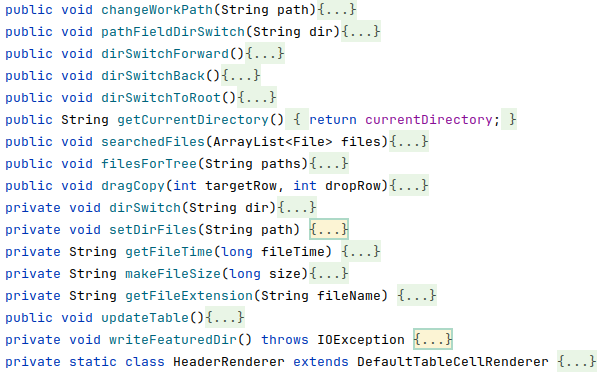


Рис 2.2 – Часть методов для отображения файлов в таблице

### 2.1.3 Конткестное меню

После успешной реализации отображения файлов и папок в таблице, следующим шагом была активация контекстного меню, позволяющего пользователям выполнять различные операции непосредственно с выбранными элементами (Рис. 2.3 – Рис. 2.4). Для этого требовалось добавить и настроить обработчики событий для контекстного меню, обеспечивая интерактивность и функциональность приложения (Рис. 2.5). Обработчики событий выполняют методы для работы с файлами.

Так же была добавлена возможность выделения файла / директории с помощью нажатия.

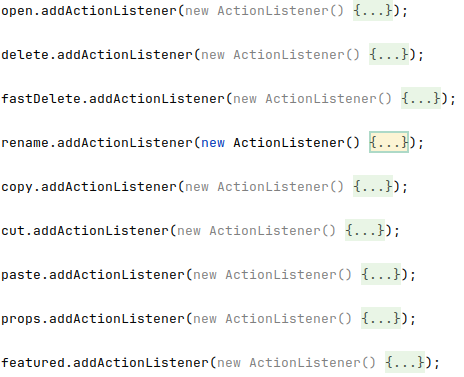


Рис 2.3 – Обработчики событий нажатия для контекстного меню 1

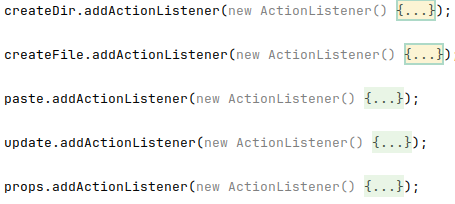


Рис 2.4 – Обработчики событий нажатия для контекстного меню 2

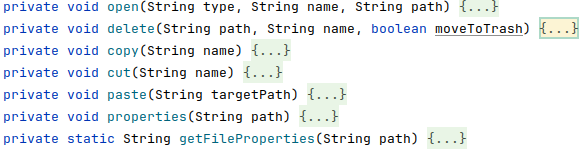


Рис 2.5 – Методы для работы с файлами / папками

### 2.1.4 Горячие клавиши

После успешной реализации контекстного меню были добавлены горячие клавиши для выполнения операций над файлами и папками, а также функционал был дополнен возможностью открытия элементов по двойному клику. Для этого были созданы обработчики событий нажатия клавиш, а так же нажатия. Эти улучшения позволили пользователям взаимодействовать с приложением более эффективно и удобно, используя как мышь, так и клавиатуру.

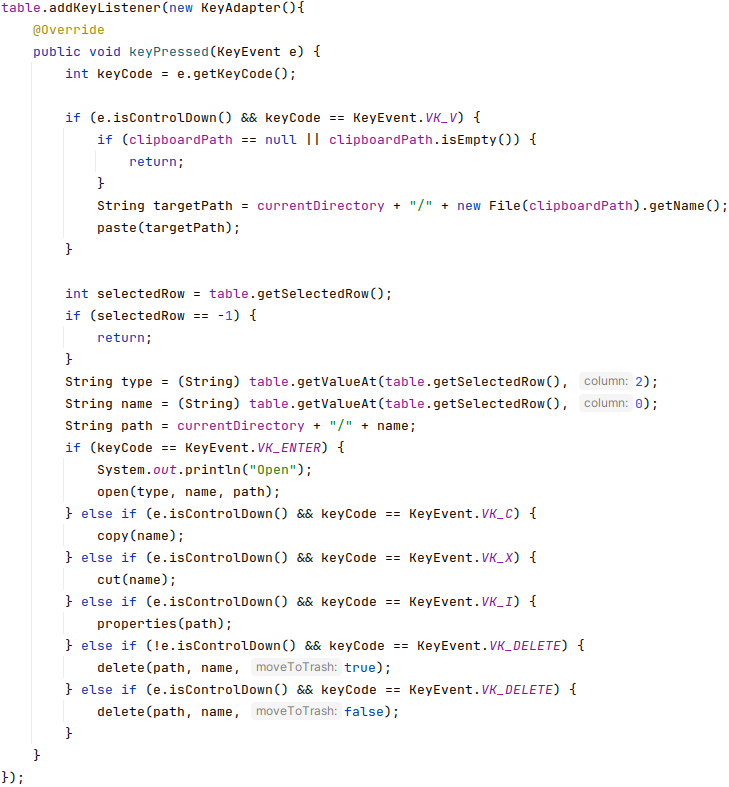
****

Рис 2.6 – Обработка горячих клавиш

### 2.1.5 Доработка таблицы с файлами

Далее была добавлена возможность сортировки в таблице (Рис 2.7).

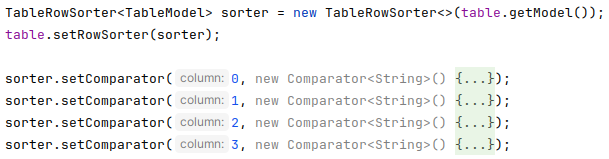


Рис 2.7 – Описание сортировки в таблице

Для более удобной навигации по директориям была добавлена кнопка в первом поле таблицы.

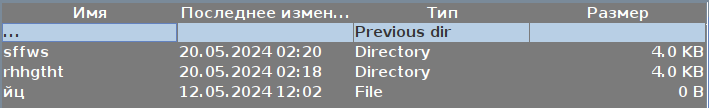


Рис 2.8 – Кнопка для возврата в прошлую директорию

В итоге была получена рабочая таблица для отображения файлов.

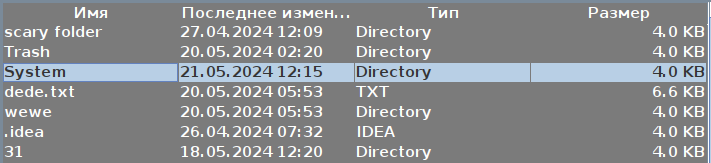


Рис 2.9 – Итоговая файловая таблица

### 2.1.6 Поисковая панель

Следующим значимым шагом в разработке приложения стало создание поисковой панели. Этот компонент предназначался для упрощения процесса навигации и поиска файлов и папок пользователем.

Первым делом была разработана панель текущего пути. Для отображения текущего пути используются методы setPathField (Рис. 2.10), который меняет текст поля, а также updatePath класса MainPage (Рис. 2.11), который вызывается в других методах, таких как открытие папки, возврат назад и т.д.



Рис 2.10 – Метод setPathField

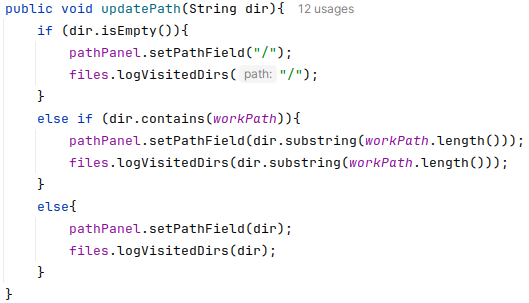


Рис 2.11 – Метод updatePath из класса MainForm

Так же необходимо было добавить возможность перехода к нужной директории через эту же панель. Для этого был использованы обработчик события этой строки (Рис 2.12).

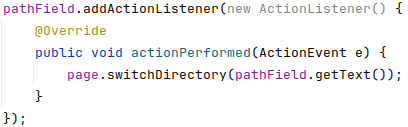


Рис 2.12 – Обработчик события pathField

Далее были реализованы кнопки навигации, которые помогают быстро переключаться по папкам. Для их работы были обработаны события нажатия на эти кнопки. При нажатии, вызывается метод dirSwitchArrows() с определенным параметром (Рис 2.13). Этот параметр определяет направление переключения директории: в предыдущую, следующую или в корневую папку.

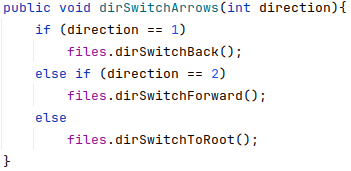


Рис 2.13 – Метод dirSwitchArrows()

Следом было решено сразу реализовать поиск по файлам. Для этого при нажатии на соответствующую кнопку появлялось окно, в которое пользователю нужно было ввести слово, по которому будет произведен поиск (Рис 2.14). Далее вызывалась функция searchFiles, которая осуществляла поиск (Рис 2.15).

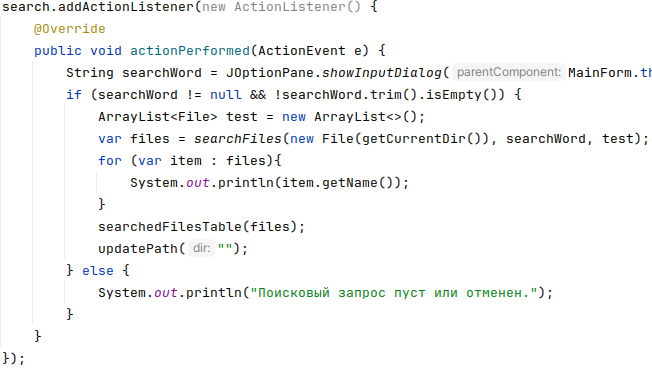


Рис 2.14 – Обработчик кнопки поиска

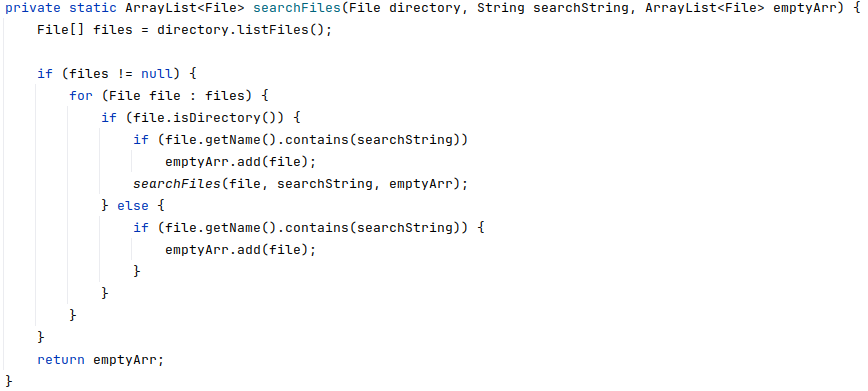


Рис 2.15 – Функция searchFiles

В итоге вся поисковая панель была готова. Затем была добавлена нижняя панель, на которой указано количество файлов в текущей директории. Для этого при открытии папки вызывался метод setFilesCount(), в параметры которого передавалось количество строк в таблице (Рис. 2.16).

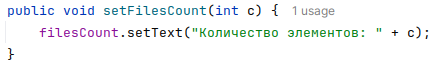


Рис 2.16 – Метод setFilesCount

### 2.1.7 Панель меню

Последним на очереди была панель меню. Для каждой кнопки был добавлен обработчик события, который выполнял соответствующее для данной кнопки действие (Рис 2.17 – Рис 2.18).

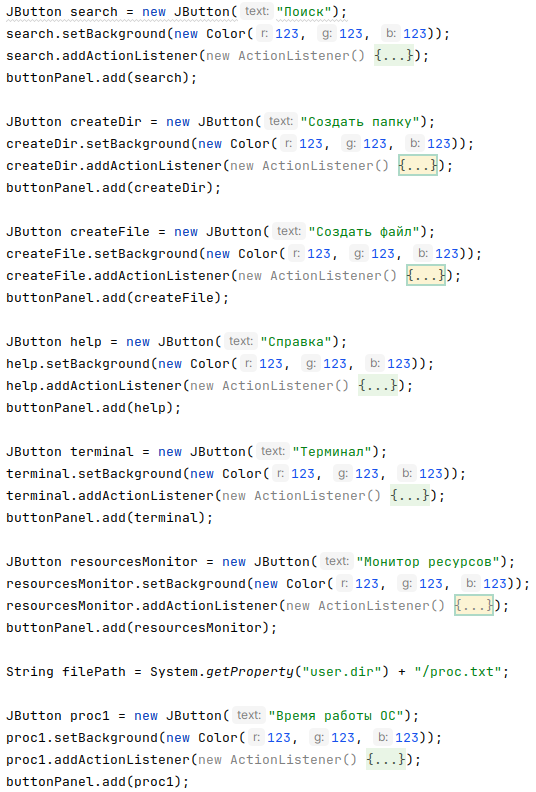


Рис 2.17 – Описание меню (1)



Рис 2.18 – Описание меню (2)

### 2.1.8 Итоги базовой версии приложения

В итоге было создано рабочее приложение SuperAPP, в котором выполняются все базовые требования курсовой работы (Рис 2.19). Следом нужно было добавить дополнительный функционал из заданий.

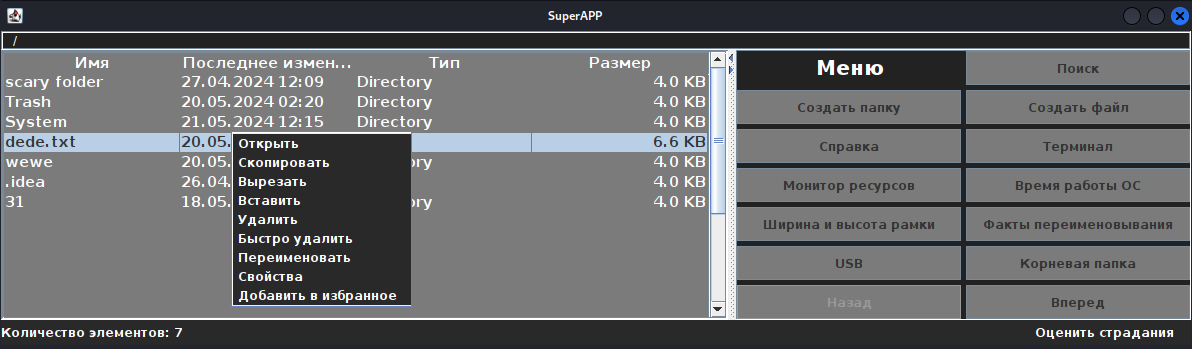


Рис 2.19 – Итоговая версия приложения

### 2.1.9 Межпроцессное взаимодействие (задание 1)

Для реализации межпроцессного взаимодействия был создан абстрактный класс FileMonitorApp, который определяет структуру для будующих процессов из 4 задания, а так же их методы для выполнения действия, отслеживания изменения файла, записи и чтения из файла (Рис 2.20).

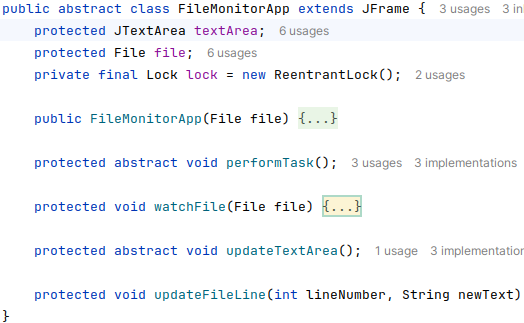


Рис 2.20 – Код класса FileMonitorApp

### 2.1.10 Отслеживание процессов (задание 2)

В следующем этапе разработки приложения была реализована функциональность для отслеживания процессов в соответствии с заданным вариантом. Эта возможность позволяла мониторить активные процессы на компьютере пользователя, и сохранять полученные данные в файл, имя которого указывалось пользователем. Для этого был создан обработчик нажатия на кнопку в меню, а так же сам метод, который отслеживал и сохранял процессы (Рис 2.21).

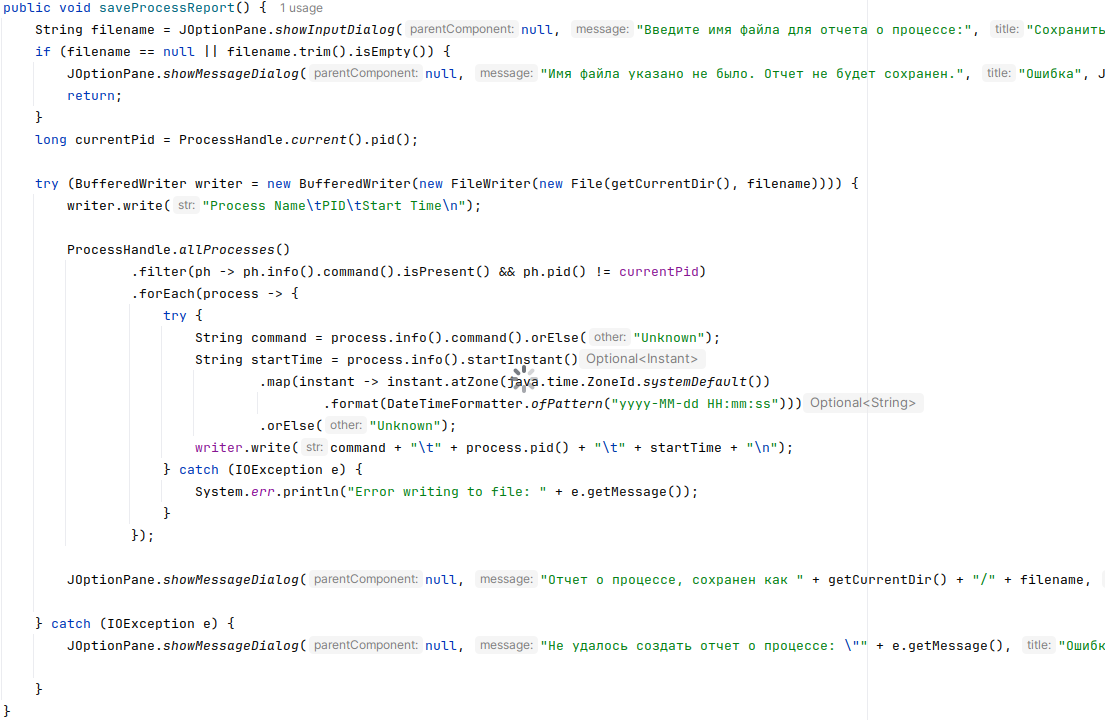


Рис 2.21 – Обработчик события для сохранения процессов в файл

### 2.1.11 Команды для терминала (задание 3)

В терминал были добавлены сетевые команды (ifconfig, route, netstat) и команды для работы с файловой системой (find, exists, size, delete, create, rename, move, copy, read, write), реализованые с помощью системных вызовов (Рис 2.22).

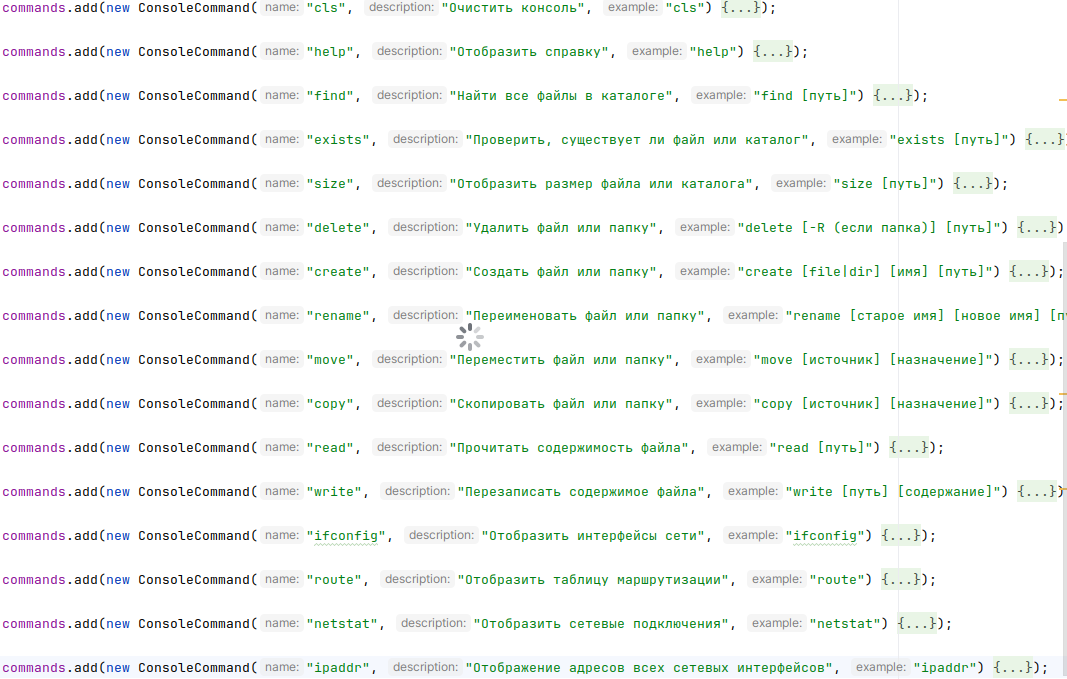


Рис 2.22 – Команды терминала

### 2.1.12 Дополнительные процессы (задание 4)

Было создано 3 класса, каждый из которых вычисляет нужную информацию, записывает их в файл, а затем считывает оттуда то, что нужно вывести на экран пользователю. Для того, что бы облегчить чтение и запись из файла, было решено сделать так, что бы каждый процесс записывал и считывал информацию только с присвоеной ему строчки.

UptimeApp записывает вывод команды uptime в первую строку файла с периодичностью в 1 секунду (Рис 2.23). Затем он ее считывает и выводит на экран (Рис 2.24).

Для WindowsSizeApp использовался getInsets(), который получает размеры рамки окна. Эта информация записывается в файл на вторую строку (Рис 2.25 – Рис 2.26).

Для обнаружения фактов переименовывания каталогов был дополнен метода rename класса FileList. При удачном выполнении переименования, вызывался метод updateLastRenameDirectory, который записывал в файл старое имя каталога (Рис 2.27). Следом DirectoryRenameApp считывал эту информацию и выводил пользователю (Рис 2.28 – Рис 2.29).

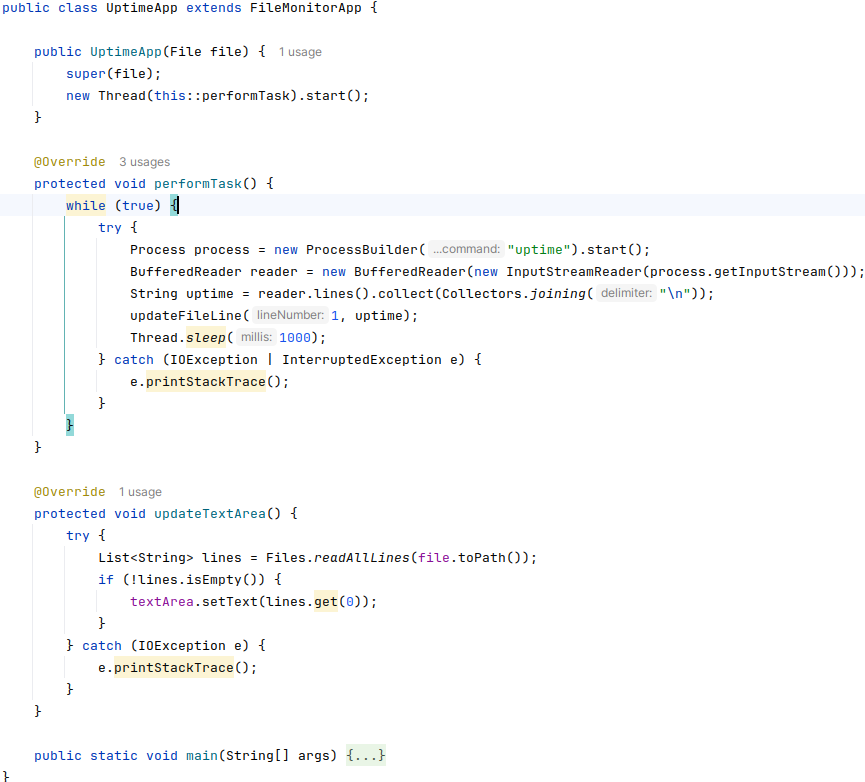


Рис 2.23 – Код класса UptimeApp

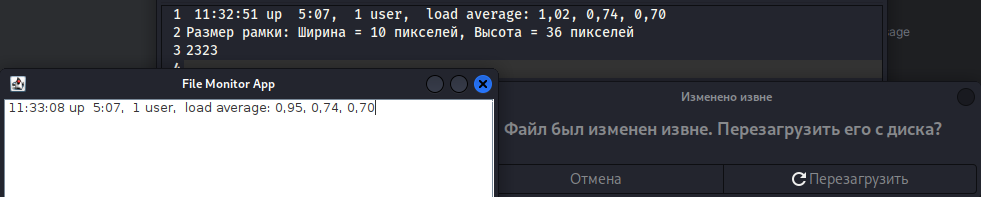


Рис 2.24 – Вывод UptimeApp



Рис 2.25 – Код класса WindowSizeApp

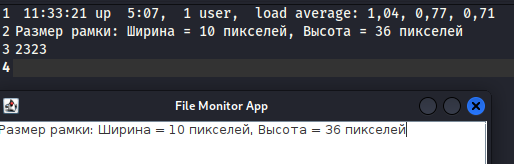


Рис 2.26 – Вывод WindowSizeApp

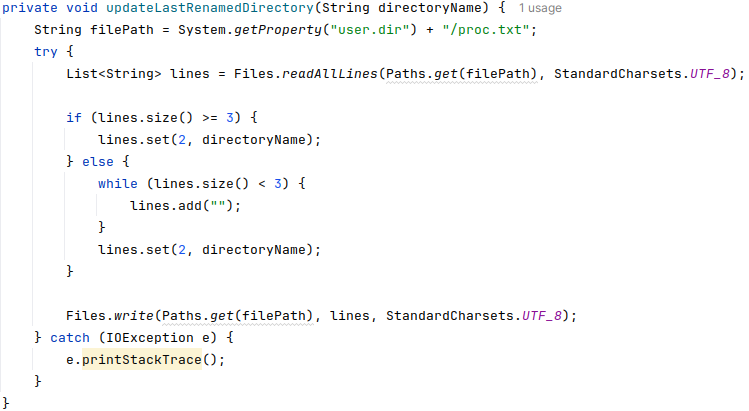


Рис 2.27 – Метода updateLastRenameDirectory



Рис 2.28 – Код класса DirectoryRenameApp

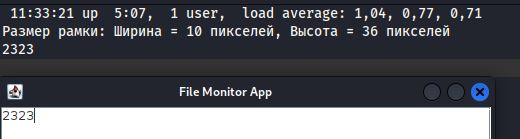


Рис 2.29 – Вывод DirectoryRenameApp

### 2.1.13 Логирование

Следующим было реализовано логирование. Метод logVisitedDirs() создавал файл visitLog.txt и обновлял его каждый раз, когда вызывался метод updateTable() (Рис 2.30).



Рис 2.30 - Метод logVisitedDirs()

### 2.1.14 Drag and Drop

Последним была реализована функция Drag And Drop (Рис 2.31). С ее помощью можно перемещать файлы и папки в другие, просто перенеся их строку в другую (Рис 2.32).



Рис 2.31 – Код Drag And Drop

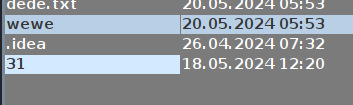


Рис 2.32 – Работа Drag And Drop

В результате этапа разработки было получено ожидаемое программное обеспечение. Так же была создана UML-диаграмма, которую можно найти в Приложении А.

## 2.2 Тестирование

Тестирование программного обеспечения - это неотъемлемая часть процесса разработки, которая позволяет оценить его качество, производительность и безопасность до выпуска. Этот процесс также помогает выявить потенциальные проблемы, снижая риск возникновения неожиданных сбоев после запуска продукта.

Была определена критическая секция программы. Несколько процессов одновременно работали с одним файлом. Что бы избежать конфликтов, в метод, который записывает данные в файл, была добавлена заглушка ReentrantLock (Рис 2.33). Заглушка помогает предотвращатить проблемы, связанные с параллельным доступом к файлу.

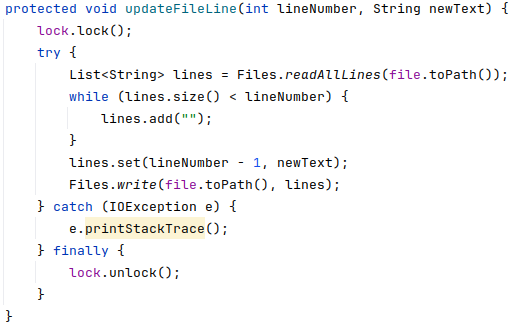


Рис 2.33 – Критическая секция

Для успешного тестирования "Супераппа" требуется разработать структурированную стратегию, выстроенную на конкретных требованиях проекта. Эта стратегия должна содержать четко сформулированный набор целей тестирования и подбор наиболее подходящих методов для их достижения.

Для проверки приложения "Суперапп" мы выбрали комбинацию методик тестирования "черного ящика" и "белого ящика". Мы сосредоточимся на тестировании основных функций программного обеспечения, таких как открытие, копирование, перемещение и удаление файлов, а также создание новых каталогов, поиск файлов и папок, работу съемных носителей, функциональность терминала приложения и возможность использования функции Drag and Drop.

Критерии тестирования для приложения "Суперапп":

1. Проверка значений исходных данных: убеждаемся, что объекты файловой системы компьютера корректно отображаются в программе "Суперапп".
2. Проверка ожидаемого результата: убеждаемся, что функции приложения "Суперапп" выполняются корректно над объектами файловой системы компьютера.
3. Проверка реакции программы: убеждаемся, что программа предупреждает пользователя о возможных ошибках при выполнении функций. Если ошибок нет, программа переходит к выполнению соответствующих функций.
4. Проверка правильности работы функций приложения с помощью проводника операционной системы: сравниваем результаты работы приложения с ожидаемыми результатами, используя файловый менеджер операционной системы.
5. Документирование возникающих ошибок и их описание: фиксируем любые обнаруженные ошибки и подробно описываем их характеристики и условия воспроизведения.

Представленные критерии обеспечат полное и надежное тестирование функциональности приложения "Суперапп". Таблица 2.1 содержит основные требования к тестированию функций приложения "Суперапп".

Таблица 2.1

Тест-требования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Входные значения | Ожидаемый результат | Конечный результат | Примечание |
| Открытие файлов | Объекты файловой системы | Открытие выбранного файла | Файл открывается | Ошибки отсутствуют |
| Копирование файлов | Объекты файловой системы | Копирование выбранного файла | Файл копируется | Ошибки отсутствуют |
| Перемещение файлов | Объекты файловой системы | Перемещение выбранного файла | Файл перемещается | Ошибки  отсутствуют |
| Удаление файлов | Объекты файловой системы | Удаление выбранного файла в корзину | Файл удаляется в корзину | Ошибки отсутствуют |
| Создание новых каталогов | Объекты файловой системы | Создание нового каталога | Создается новый каталог | Ошибки отсутствуют |
| Поиск файлов и папок | Объекты файловой системы | Поиск файла по имени, вывод списка совпадений | Выводится список совпадений | Ошибки отсутствуют |
| Работа со съемными носителями | Подключаемые внешние накопители | Отображение нового съемного носителя | Отображается новый съемный носитель | Ошибки отсутствуют |
| Работоспособность терминала приложения | Вводимые команды пользователем в терминал | Отображение результата ввода команды | Отображается результат ввода команды | Ошибки отсутствуют |
| Drag and Drop | Объекты файловой системы | Выполнение команды Drag and Drop | Происходит перемещение / копирование | Ошибки отсутствуют |

Результаты тестирования приложения «СуперАпп» представлены в табл. 2.2 и табл. 2.3.

Таблица 2.2

Результаты тестирования приложения

| Функции прилож.  № | Открытие файлов | Копирование файлов | Перемещение файлов | Удаление файлов | Создание новых каталогов |
| --- | --- | --- | --- | --- | --- |
| 1. | + | + | + | + | + |
| 2. | + | + | + | + | + |
| 3. | + | + | + | + | + |
| 4. | + | + | + | + | + |
| 5. | - | - | +  (ошибка возникает, если пользователь пытается переместить элементы корневой папки System) | +  (ошибка возникает, если пользователь пытается удалить элементы корневой папки System, которую нельзя изменить) | - |

Таблица 2.3

Результаты тестирования приложения

| Функции приложения  № | Поиск файлов и папок | Работа со съемными носителями | Работоспособность терминала приложения | Drag and Drop |
| --- | --- | --- | --- | --- |
| 1. | + | + | + | + |
| 2. | + | + | + | + |
| 3. | + | + | + | + |
| 4. | + | + | + | + |
| 5. | - | - | - | - |

# 3 Руководство пользователя

## 3.1 Введение

Добро пожаловать в пользовательское руководство программы SuperApp! Это руководство создано для тех, кто желает узнать об основах работы с нашим программным продуктом и извлечь максимум пользы из его возможностей.

В данном руководстве представлены подробные инструкции по установке и запуску SuperApp, а также рекомендации по использованию всех ключевых функций программы. Мы постарались сделать инструкции максимально ясными и доступными, чтобы вы могли легко начать работу и эффективно использовать приложение в своих ежедневных задачах.

Приятного использования!

## 3.2 Назначение и условия применения

SuperApp разработан для разнообразной аудитории пользователей, которым необходимо эффективно управлять файлами. Эта программа подходит как для использования в домашних условиях, так и в профессиональной среде для повышения эффективности обработки данных и оптимизации использования ресурсов компьютера. SuperApp не предъявляет особых требований к условиям эксплуатации и может быть установлена на любой системе, которая поддерживает Linux.

Минимальные системные требования для приложения:

1. ОС Linux
2. Процессор Intel Celeron 1,6-1,8GHz.
3. Оперативная память: 2 ГБ.
4. Свободное место на жестком диске: 100 МБ.
5. Видеокарта: Интегрированная в процессор.
6. Интернет соединение не требуется.
7. Звуковая карта не требуется.

## 3.3 Подготовка к работе.

Чтобы начать использование SuperApp, следуйте этим шагам:

1. **Скачивание архива**: Перейдите на официальный сайт программы или страницу загрузки и скачайте последнюю версию программы в виде архива.
2. **Распаковка архива**: После загрузки файла архива сохраните его в удобное для вас место на вашем компьютере. **ВАЖНО!!!** **Программа будет работать только с теми файлами и директориями, которые находятся в той же папке, где находится папка «System» данной программы.**
3. **Установка программы**: Откройте терминал. Переместитесь в директорию, куда вы распаковали архив, используя команду cd, например: cd /путь/к/папке/System.

**ВАЖНО!!! Для первого запуска программы запустить скрипт инициализации с правами администратора (root), используя команду: sudo ./init.sh.** Эта команда запустит процесс настройки и запуска программы.

Для открытия программы в дальнейшем, можете так же переходить в директорию с файлом init.sh, и запускать его, но теперь уже без прав администратора, используя команду: ./init.sh

После выполнения этих шагов вы готовы начать работу с SuperApp и исследовать функционал, предоставляемый программой. Убедитесь, что вы следовали всем инструкциям корректно, чтобы избежать возможных ошибок в работе программы.

## Описание операций.

SuperApp предоставляет ряд функций для управления файлами, мониторинга системных ресурсов и работы с устройствами ввода/вывода. Ниже представлено подробное описание основных операций, доступных в программе:

* **Создание файлов и папок**: Для этого можно использовать как меню «File» панели инструментов, так и контекстное меню, нажав ПКМ по пустому пространству в таблице с файлами.
* **Копирование и перемещение**: Для копирования и перемещения нужно выделить файл / директорию, который хотим переместить, вызвать контекстное меню и выбрать в нем нужную функцию. После чего перейти в директорию, в которую хотим переместить файл / директорию, нажать ПКМ по пустой области и выбрать функцию Paste. Так для этого можно использовать горячие клавиши, такие как: Ctrl + C для копирования, Ctrl + X для перемещения и Ctrl + V для вставки. Подробнее о горячих клавишах можно узнать в «File – Help».
* **Удаление**: Файлы и папки можно удалять напрямую или перемещать в корзину. Для этого используются соответствующие кнопки в контекстном меню, или горячие клавиши: del для перемещения в корзину и Ctrl + del для быстрого удаления без перемещения в корзину.
* **Поиск**: Поиск файлов осуществляется по имени, обеспечивая быстрый доступ к нужным файлам. Для этого в поисковую строку надо ввести название файла, который мы хотим найти. После этого в таблице появятся все нужные нам файлы.
* **Переименование**: Для этого нужно выделить нужный нам файл / директорию, вызвать контекстное меню и выбрать в нем «Rename».
* **Получение свойств**: Для получения свойств нужно выделить нужный нам файл / директорию, вызвать контекстное меню и выбрать в нем «Properties», или нажать Ctrl + I.
* **Избранное**: В SuperAPP так же есть возможность добавления директорий в левое меню в папку Features, а так же удаления от туда. Для этого нужно выделить директорию, вызвать контекстное меню и выбрать в нем «Add to featured», после чего к этой директории можно быстро перейти, используя папку «Featured» в левой панели программы. Для удаления оттуда, нужно выделить директорию в левой панели и нажать «Remove from featured».
* **Терминал**: В программе так же есть возможность использования встроенного терминала, для быстрых операций с файлами. Для этого нужно запустить его из «Utils - Terminal». Что бы ознакомиться с его командами, наберите в нижней строке терминала команду help.
* **Монитор ресурсов**: Из SuperAPP можно вызывать монитор ресурсов.

## Аварийные ситуации.

В этом разделе руководства пользователя мы обсудим потенциальные аварийные ситуации, которые могут возникнуть при использовании программы SuperApp, а также предоставим рекомендации по их предотвращению и устранению. Понимание этих сценариев и знание эффективных методов реагирования поможет вам минимизировать риск потери данных и обеспечить стабильность работы приложения.

1. Зависание программы

Описание: Зависание программы может произойти из-за ошибок в программе или неправильного взаимодействия с другими приложениями.

Рекомендации:

* Закройте и перезапустите приложение.
* Проверьте наличие обновлений для SuperApp и установите их.

1. Отсутствие системных файлов программы

Описание: Приложение SuperApp может перестать функционировать должным образом или вообще не запуститься, если в его системной папке отсутствуют необходимые файлы. Это может произойти из-за их случайного удаления, повреждения в результате сбоев оборудования, вирусной атаки или ошибочных действий пользователя.

Рекомендации:

* Восстановление файлов: Из резервной копии: Если регулярно создаются резервные копии системных файлов, восстановите недостающие файлы из последней актуальной копии.
* Переустановка приложения: Если резервные копии отсутствуют, рекомендуется полностью переустановить приложение для восстановления всех системных файлов в их первоначальном состоянии.
* Обращение в службу поддержки: Если проблему не удается решить самостоятельно, следует по адресу [supporting.superapp@gmail.com](mailto:supporting.superapp@gmail.com) для получения помощи и дополнительных инструкций.

## Рекомендации по освоению.

Для того чтобы максимально эффективно использовать возможности SuperApp и обеспечить плавный процесс освоения программы, мы предлагаем следующие рекомендации:

1. **Изучение интерфейса пользователя**

* **Ознакомьтесь с интерфейсом**: Проведите некоторое время, исследуя интерфейс SuperApp, чтобы понять расположение основных элементов управления и доступные функции.

1. **Практическое применение**

* **Начните с базовых операций**: Попробуйте выполнить простые задачи, такие как копирование, перемещение и удаление файлов, чтобы получить представление о функциональности программы.
* **Переход к сложным задачам**: По мере роста уверенности начните использовать более сложные функции, такие как терминал.

1. **Регулярное использование**

* **Интегрируйте SuperApp в ежедневные задачи**: Чем чаще вы будете использовать программу, тем лучше узнаете ее возможности и тем эффективнее она станет инструментом в вашей работе.
* **Используйте программу для решения реальных задач**: Применяйте SuperApp для организации файлов, управления данными и мониторинга системы в реальных проектах.

1. **Обратная связь и поддержка**

* **Используйте техническую поддержку**: Не стесняйтесь обращаться в службу поддержки за помощью или пояснениями, если столкнетесь с трудностями. Наш адрес электронной почты: [supporting.superapp@gmail.com](mailto:supporting.superapp@gmail.com)

Следуя этим рекомендациям, вы сможете не только эффективно использовать SuperAPP для своих нужд, но и значительно улучшить свои навыки работы с компьютером и программным обеспечением.

# Заключение

В ходе выполнения этой курсовой работы были успешно достигнуты установленные цели и задачи, включающие углубление теоретических знаний и приобретение практических навыков в дизайне пользовательских интерфейсов и разработке программного обеспечения для взаимодействия с файлами.

Проект эффективно показал, как можно применять знания, полученные во время лекций и на практических занятиях. Было создано программное обеспечение, отвечающее всем современным стандартам программирования, что подтверждает высокий уровень приобретённых профессиональных умений.

Особое внимание было уделено разработке графического интерфейса с использованием Figma, что облегчило реализацию функциональных требований к программе и сделало её удобной и понятной для пользователя.

Тестирование разработанного программного продукта подтвердило его функциональность и стабильность, что является критически важным для профессиональной разработки программного обеспечения. Созданное приложение не только соответствует всем требованиям, но также демонстрирует возможности современных операционных систем в управлении ресурсами и межпроцессном взаимодействии.

В процессе работы над курсовой работой были развиты значимые умения, охватывающие не только технические аспекты создания программ, но и организационные и исследовательские навыки. Возможность применять теоретические знания в реальном проекте предоставила бесценный опыт, который будет способствовать дальнейшему профессиональному росту в области информационных технологий.

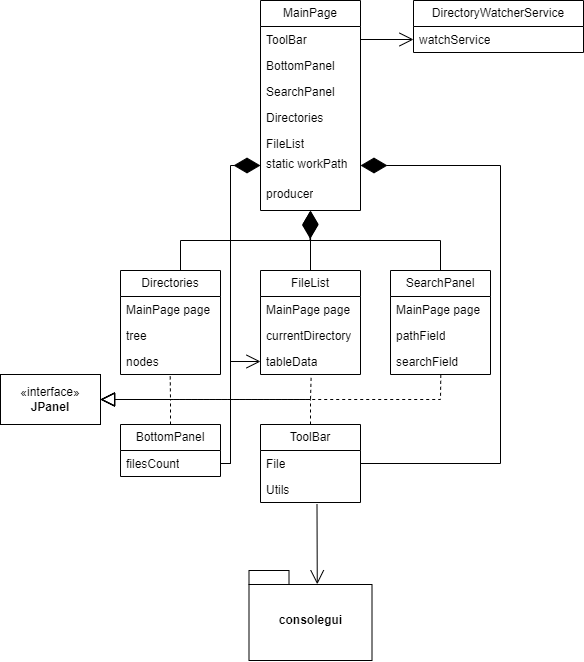
Таким образом, реализация этой курсовой работы не только позволила закрепить полученные знания и умения, но и заложила основу для будущей профессиональной деятельности, что неминуемо поможет в строительстве успешной карьеры в разработке и управлении системным программным обеспечением.

# Список использованных источников

1. Гудвин, Г.К. Проектирование программного обеспечения [Текст] / Г.К. Гудвин, С.Ф. Гребе, М.Э. Сальдаго. – пер. с англ. – М.: БИНОМ, Лаборатория знаний, 2004. – 911 с.
2. Брюханов, В.Н. Теория управления программным обеспечнием [Текст] : Учеб. для машиностроит. спец. вузов / В.Н. Брюханов, М.Г. Косов, С.П. Протопопов и др.; под ред. Ю.М. Соломенцева – 3-е изд., стер. – М.: Высш. шк., 2000. – 268 с
3. Руководство по Fibre Channel [Электронный ресурс] // Мир компьютерной автоматизации. -2021. -Режим доступа: http://www.mka.ru/?p=40030/, свободный. -Загл. с экрана.

# Приложение А

**UML-диаграмма**

****