

УТВЕРЖДАЮ

Руководитель практики от Университета

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

*(должность, ФИО)*

МП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(Подпись)*

Дата « \_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 202\_ год

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ.05 Проектирование и разработка информационных систем

|  |  |
| --- | --- |
| Щекотова Екатерина Алексеевна | |
| *(Ф.И.О. обучающегося)* | |
| 09.02.07 Информационные системы и программирование | |
| *(специальность)* | |
|  | |
| Учебная группа | ИСПк-203-52-00 |
|  |  |
| Место прохождения практики | ФГБОУ ВО «Вятский государственный университет», |
| Колледж ВятГУ | *(наименование организации, структурного подразделения организации)* |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Итоговая оценка: |  | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | *(дата)* |  | *(подпись)* |  | *(Ф.И.О.)* |

Киров, 2023 г.



ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | | | Щекотова Екатерина Алексеевна | | | | | | |
| Специальность | | | 09.02.07 Информационные системы и программирование | | | | | | |
| Учебная группа | | | ИСПк-203-52-00 | | | | | | |
| Вид практики | | | учебная практика | | | | | | |
| Сроки прохождения практики с | | | | 09.01.2023 | по | 15.06.2023 | | | |
| Место прохождения практики | | | | ФГБОУ ВО «Вятский государственный университет»,  Колледж ВятГУ | | | | | |
|  | | (наименование организации, структурного подразделения организации) | | | | | | |
| № | Виды работ, выполняемых обучающимися во время практики | | | | | | Объем работ (час) | Формируемые компетенции | | |
| 1 | Пройти инструктаж по ознакомлению с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте | | | | | | 2 | - | | |
| 2 | Постановка задачи | | | | | | 8 | ОК2, ОК6, ОК7, ОК9 | | |
| 3 | Настройка рабочего окружения | | | | | | 8 | ОК1, ОК8, ОК9 | | |
| 4 | Анализ предметной области, постановка расширенного технического задания | | | | | | 10 | ПК5.1, ОК3, ОК4 | | |
| 5 | Проектирование | | | | | | 35 | ПК5.2, ПК5.3, | | |
| 6 | Реализация программного кода | | | | | | 35 | ПК5.4, ПК5.8, | | |
| 7 | Тестирование и отладка полученного кода | | | | | | 22 | ПК5.5, ПК6.4, ОК2, ОК9 | | |
| 8 | Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов | | | | | | 10 | ПК5.6, ПК5.7, ОК9, ОК10, ОК11 | | |
| 9 | Подготовка к сдаче и сдача промежуточной аттестации | | | | | | 2 | ОК4, ОК5 | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Индивидуальное задание на практику разработано в соответствии с рабочей программой практики. | | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | (дата) |  | (подпись) |  | (Ф.И.О.) |

|  |  |
| --- | --- |
| С индивидуальным заданием ознакомлен(а) |  |
|  | (дата, подпись обучающегося) |

**ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | | Щекотова Екатерина Алексеевна | | | |
| Специальность | | 09.02.07 Информационные системы и программирование | | | |
| Учебная группа | | ИСПк-203-52-00 | | | |
| Вид практики | | учебная практика | | | |
| Сроки прохождения практики с | | | 09.01.2023 | по | 15.06.2023 |
| Место прохождения практики | | | ФГБОУ ВО «Вятский государственный университет», Колледж ВятГУ | | |
|  | (наименование организации, структурного подразделения организации) | | | | |

ВИДЫ И КАЧЕСТВО ВЫПОЛНЕНИЯ РАБОТ

|  |  |  |  |
| --- | --- | --- | --- |
| Вид работ | Критерий выполнения работ | | |
| Выполнены полностью самостоятельно | Выполнены с незначительной помощью наставника | Выполнены с помощью наставника |
| Постановка задачи | V |  |  |
| Настройка рабочего окружения | V |  |  |
| Анализ предметной области, постановка расширенного технического задания | V |  |  |
| Проектирование | V |  |  |
| Реализация программного кода | V |  |  |
| Тестирование и отладка полученного кода | V |  |  |
| Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов | V |  |  |
| Подготовка к сдаче и сдача промежуточной аттестации | V |  |  |

Обучающийся ознакомлен с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также прошел вводный инструктаж и инструктаж на рабочем месте.

Во время прохождения учебной практики обучающимся освоены следующие профессиональные и общие компетенции:

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование компетенции | Показатели оценки | Оценка | |
| Освоена | Не освоена |
| ПК 5.1. Собирать исходные данные для разработки проектной документации на информационную систему. | Способен выполнять анализ предметной области, выявляя существенные элементы, оказывающие влияние на проектируемую систему | V |  |
| ПК 5.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика. | Способен выполнять декомпозицию сущностей с целью получения наиболее полной картины о целесообразной структуре разработки | V |  |
| ПК 5.3. Разрабатывать подсистемы безопасной информационной системы в соответствии с техническим заданием. | Способен прогнозировать потенциально некорректные действия пользователя и предусматривать соответствующие реакции со стороны системы | V |  |
| ПК 5.4. Производить разработку модулей информационной системы в соответствии с техническим заданием. | Способен создавать программный код, отвечающий предъявляемым требованиям | V |  |
| ПК 5.5. Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы. | Способен выполнять оценку корректности функционирования системы | V |  |
| ПК 5.6. Разрабатывать техническую документацию на эксплуатацию информационной системы. | Способен разрабатывать проектную, техническую и пользовательскую документации | V |  |
| ПК 5.7. Производить оценку информационной системы для выявления возможности ее модернизации. | Способен оценивать перспективы дальнейшего развития программной системы | V |  |
| ПК 5.8. Осуществлять интеграцию информационной системы с другими программными продуктами. | Способен повторно использовать готовые шаблонные решения при разработке программного продукта | V |  |
| ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. | Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности | V |  |
| ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. | Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности | V |  |
| ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие. | Способен находить пути улучшения имеющихся решений, позволяющих повысить их общий качественный уровень | V |  |
| ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. | Способен эффективно использовать компьютерное время, а также материальные ресурсы, необходимые для решения поставленных задач | V |  |
| ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. | Способен соблюдать требования внутреннего трудового распорядка организации, охраны труда и техники безопасности в целях сохранения собственного здоровья | V |  |
| ОК 09. Использовать информационные технологии в профессиональной деятельности. | Способен применять стек современных средств разработки ПО для решения задач профессиональной деятельности в заданном контексте | V |  |
| ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке. | Способен использовать в своей работе специализированную документацию | V |  |
| ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере. | Способен проектировать алгоритмические решения, принимая во внимание имеющиеся ресурсные ограничения | V |  |

**Краткая характеристика работы обучающегося**

|  |
| --- |
| Программа практики выполнена в полном объеме. Все виды работ выполнялись в срок, |
| без существенных замечаний. В достаточной степени была проявлена самостоятельность |
| и умение грамотно пользоваться источниками информации, находящимися в свободном |
| доступе. |

|  |  |  |
| --- | --- | --- |
|  |  | Руководитель практики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Подпись ФИО  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность)  Дата «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_\_ год |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc146249507)

[1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ 6](#_Toc146249508)

[2. НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ 7](#_Toc146249509)

[3. ФОРМУЛИРОВКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ 9](#_Toc146249510)

[4. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ 10](#_Toc146249511)

[4.1 Анализ предметной области и обзор аналогов 10](#_Toc146249512)

[4.2 Разработка структуры приложения и алгоритмов функционирования. 15](#_Toc146249513)

[4.3 Реализация программы. Разработка пользовательского интерфейса 23](#_Toc146249514)

[4.4 Тестирование программы. 35](#_Toc146249515)

[ЗАКЛЮЧЕНИЕ 38](#_Toc146249516)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 39](#_Toc146249517)

[ПРИЛОЖЕНИЕ А 40](#_Toc146249518)

[ПРИЛОЖЕНИЕ Б 45](#_Toc146249519)

[ПРИЛОЖЕНИЕ В 46](#_Toc146249520)

[ПРИЛОЖЕНИЕ Г 47](#_Toc146249521)

# ВВЕДЕНИЕ

Учебная практика ПМ.05 проходила на базе Колледжа ФГБОУ ВО «Вятский государственный университет» в период с 09.01.2023 г. по 15.06.2023 г. по субботам.

Цель практики: сформировать у обучающихся представление о процессе разработки прикладного программного обеспечения.

Задачи практики:

– дать представление о различных методах проектирования прикладного программного обеспечения;

– сформировать навыки написания документации, требуемой при разработке прикладного программного обеспечения;

– дать представление о различных инструментах, используемых при разработке прикладного программного обеспечения.

# 1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ

В период с 09.01.2023 по 14.06.2023 при прохождении учебной практики ПМ.05 на базе Колледжа ФГОБУ ВО «Вятский государственный университет» был выполнен следующий перечень работ, представленный в таблице 1.

Таблица 1 – Сведения о работе, выполненной в период практики

| Дата | Краткое содержание выполненных работ |
| --- | --- |
| 11.01.2021 | Ознакомление с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте |
| 12.01.2021-25.01.2021 | Постановка задачи |
| 26.01.2021-08.02.2021 | Настройка рабочего окружения |
| 09.02.2021-22.02.2021 | Анализ предметной области, постановка расширенного технического задания |
| 23.02.2021-29.03.2021 | Проектирование |
| 30.03.2021-03.05.2021 | Реализация программного кода |
| 04.05.2021-24.05.2021 | Тестирование и отладка полученного кода |
| 25.05.2021-13.06.2021 | Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов |
| 14.06.2021-16.06.2021 | Подготовка к сдаче и сдача промежуточной аттестации |

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

(дата) (подпись)

# 2. НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ

В работе использовался язык Lazarus IDE. Lazarus – свободно распространяемая IDE для компилятора Free Pascal Compiler. Среда разработки Lazarus является бесплатным аналогом Delphi. Содержит набор визуальных компонентов LCL (Lazarus Component Library).

Основное преимущество Lazarus – это способность создавать исполняемые программы для различных операционных систем. В отличие от Java, .Net и т.п. Lazarus создаёт не промежуточный, а машинный код. Такой код не нуждается в интерпретаторе и работает в десять раз быстрее, чем промежуточный код. Основное преимущество Lazarus так же является и его основным недостатком. Так как программы на Lazarus переводятся в машинный код, то необходимо каждый раз компилировать программу для различных операционных систем.

Lazarus состоит из нескольких окон (которые стоит подравнять, чтобы они занимали весь рабочий стол и не мешали друг другу):

1. Главное окно;
2. Инспектор объектов;
3. Редактор форм;
4. Редактор исходного кода;
5. Окно сообщений.

Интерфейс языка программирования Lazarus представлен на Рисунке 1.

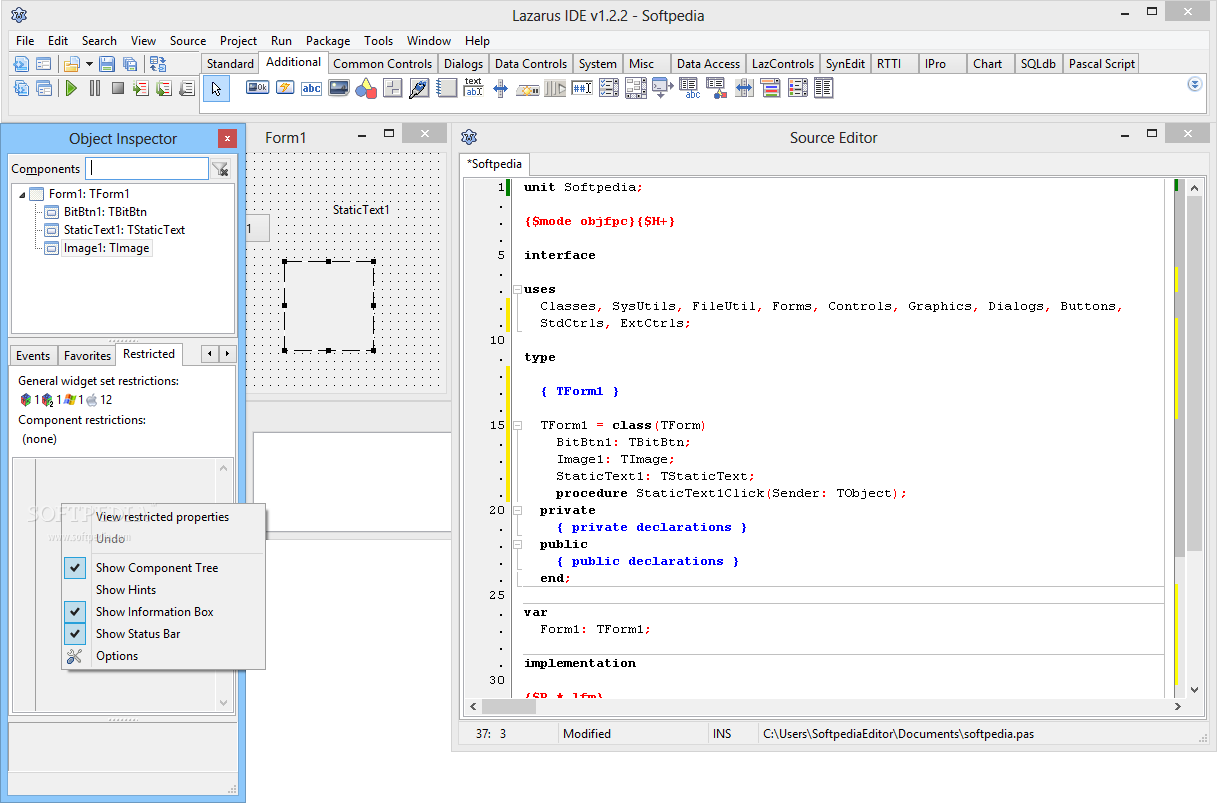


Рисунок 1 – Интерфейс Lazarus

# 3. ФОРМУЛИРОВКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

В соответствии с индивидуальным заданием на учебную практику необходимо создать проект, используя любой язык программирования, описать разрабатываемое приложение, требования к функциям, внешнему виду, структурам данных, стадиям и этапам разработки. Реализовано приложение «Блокнот».

Разработанное приложение «Блокнот» является вспомогательным средством для ведения ежедневных записей в повседневной жизни человека. С помощью него пользователь может легко и доступно составлять списки покупок, планировать бюджет, записывать важные даты или яркие впечатления из жизни.

# 4. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

## 4.1 Анализ предметной области и обзор аналогов

Блокнот – одна из самых простых программ по работе с текстом. В ней мы можем вводить текст, редактировать его и сохранять.

У электронных блокнотов действительно не счесть преимуществ по сравнению с традиционными бумажными. Они доступны на любых устройствах — от мобильных гаджетов до персональных компьютеров, всегда под рукой и одинаково хорошо подходят как для текстовых записей, так и составления списков дел, записи сиюминутных мыслей и свежих идей. С их помощью можно удобно упорядочивать и структурировать информацию, быстро находить нужные заметки, планировать задачи и объединять все записи в одном приложении.

Приложение «Блокнот» – приложение для создания, сортировки и редактирования заметок. Помимо написания текстовых записок, поддерживается изменение шрифта и размера отдельных символов, возможность сохранять и открывать сохраненные записи.

На данный момент существуют следующие аналоги приложения «Блокнот»:

Аналог №1

Программа «WordPad».

Ссылка :<https://wordpad.ru/>

Интерфейс приложения представлен на Рисунке 2.

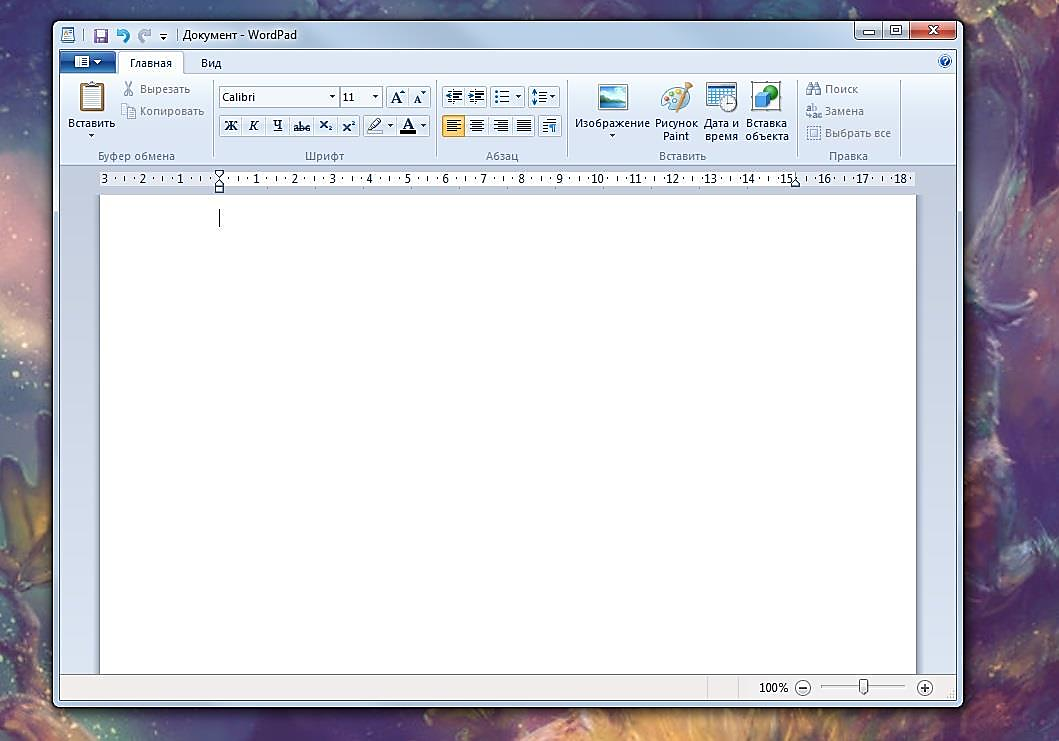


Рисунок 2 – Программа «WordPad»

**К достоинствам рассматриваемой программы можно отнести:**

* + - Не требует быть продвинутым пользователем
    - Ничего не отвлекает
    - Можно форматировать текст, делая его визуально удобным для работы и выделяя проблемные места

**К недостаткам программы относятся:**

* + - Ничего большего от программы получить не выйдет
    - Нет статистики по тексту и из размера файла её тоже не достать.
    - Нет автосохранения.
    - Нет возможности изменить дизайн под себя;

Аналог №2

Приложение «Evernote»

Ссылка: <https://evernote.com/ru>

Интерфейс приложения представлен на Рисунке 3.

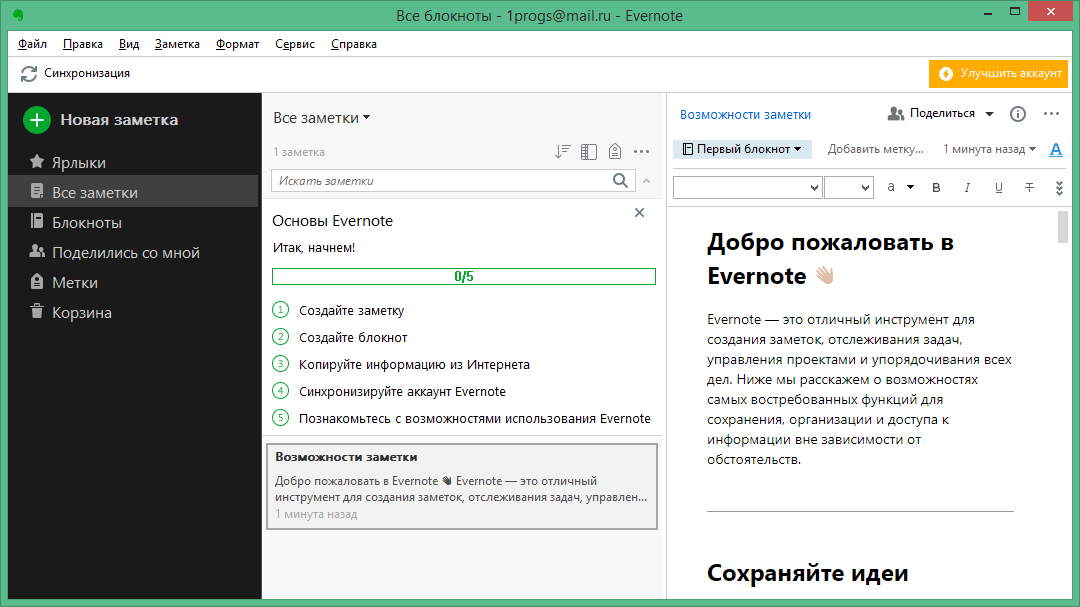


Рисунок 3 – Приложение «Evernote»

**К достоинствам рассматриваемой программы можно отнести:**

* информация хранится в удобном формате;
* все данные упорядочены, их можно структурировать;
* у сервиса сильный поиск, отыскать материал очень легко;
* можно ставить метки.

**К недостаткам программы относятся:**

* только два уровня вложенности;
* разные версии для мобильных платформ;
* занимает много памяти.

Аналог №3

Приложение «Simplenote»

Ссылка: <https://simplenote.com/>

Интерфейс предоставлен на Рисунке 4.

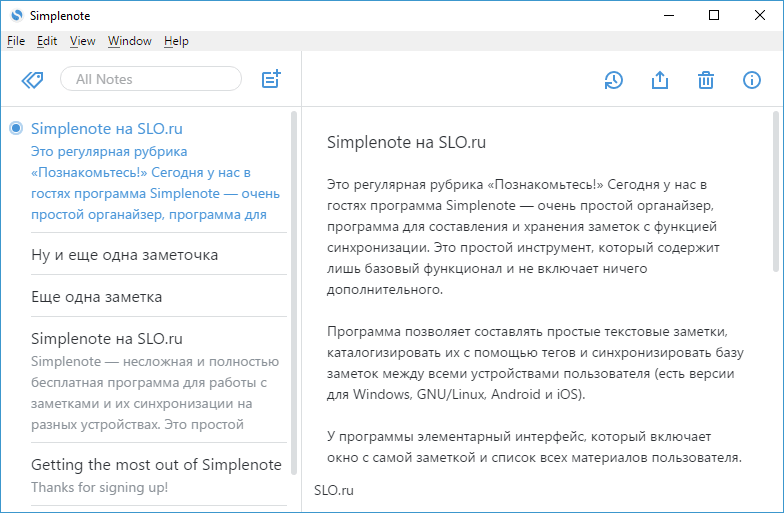


Рисунок 4 – Приложение «Simplenote»

**К достоинствам рассматриваемой программы можно отнести:**

* Симпатичный пользовательские интерфейсы, к которому быстро привыкаешь
* Поддержка на любых устройствах, включая смартфоны
* Сохранение заметок в облаке и доступ к ним с любого компьютера, ноутбука или телефона
* Низкие системные требования и отличное быстродействие

**К недостаткам программы относятся:**

* чувствительный к регистру поиск и невозможность открыть к заметке общий доступ для любого пользователя через сам Android клиент.

Аналог №4

Приложение «WorkFlowy».

Ссылка: <https://workflowy.com/>

Интерфейс предоставлен на Рисунке 5.

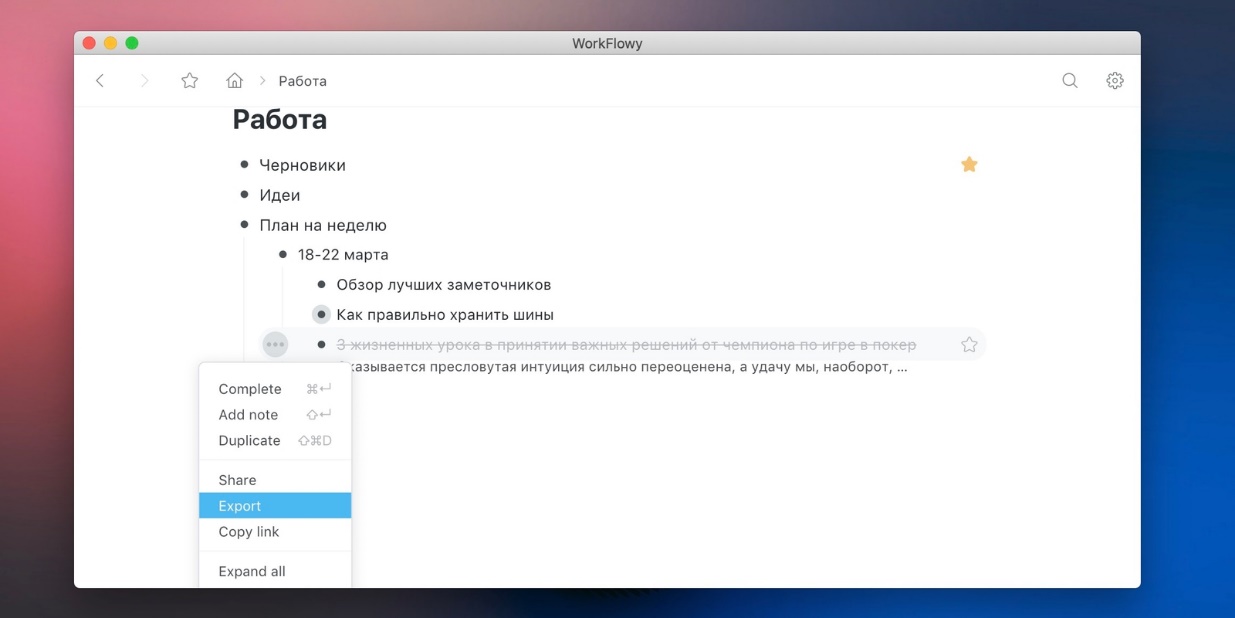


Рисунок 5 – Приложение «WorkFlowy»

**К достоинствам рассматриваемой программы можно отнести:**

* Интуитивно понятный интерфейс
* Синхронизируется в облако
* Есть версия для ПК, Android и iOS

**К недостаткам программы относятся:**

* Нельзя добавлять изображения и файлы

## 4.2 Разработка структуры приложения и алгоритмов функционирования.

Простой текстовый редактор с базовыми функциями, такими как открытие, сохранение и редактирование текста. Когда пользователь взаимодействует с элементами интерфейса, соответствующие действия выполняются на основе выбора в меню или других действий пользователя.

Программа состоит из следующих методов: инициализация программы, создание или открытие файла, ввод и редактирование текста, настройки и опции, сохранение текста в файл. (Рисунок 6).

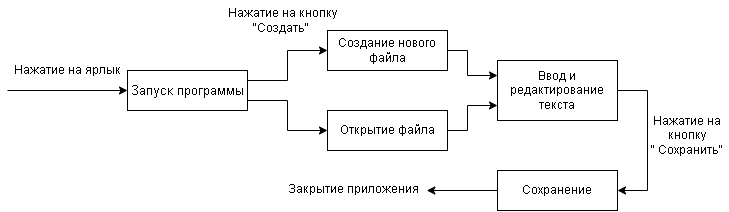


Рисунок 6 – Структура программы.

Схема алгоритма создания и открытия файла (Рисунок 7,8).

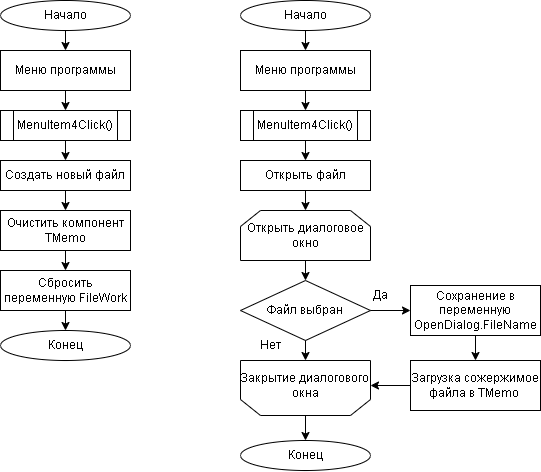


Рисунок 7 – Алгоритм создания файла.

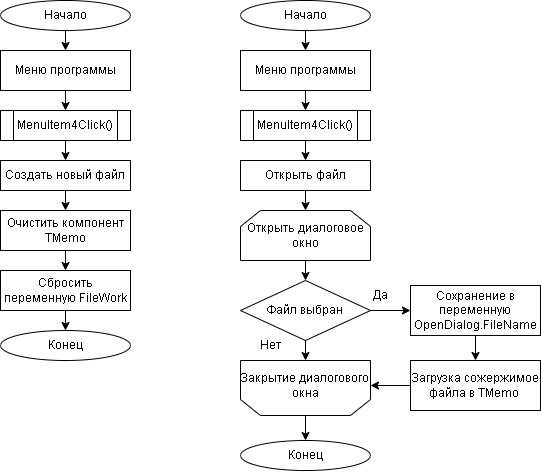


Рисунок 8 – Алгоритм открытия файла.

Таким образом, алгоритм создания и открытия файла осуществляется посредством вызова соответствующих методов обработчиков событий. Создание файла очищает компонент TMemo для начала работы с новым файлом, а открытие файла загружает содержимое существующего файла в TMemo для последующей обработки.

Пользователь может создать новый файл, используя MenuItem4Click (создать файл). Также может открыть существующий файл с текстом, используя MenuItem5Click (открыть файл). Это событие загружает содержимое файла в компонент Memo1.

Алгоритм ввода и редактирования текста (Рисунок 9).

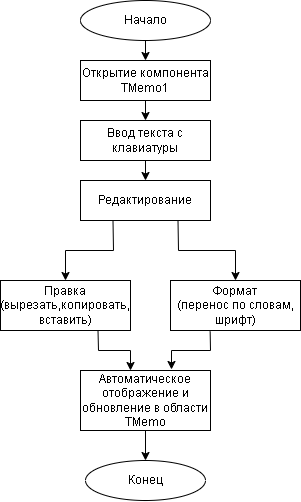


Рисунок 9 – Алгоритм ввода и редактирования теста.

Алгоритм ввода текста в данном редакторе представлен с использованием компонента TMemo, который обрабатывает ввод текста автоматически. При использовании TMemo пользователи могут вводить текст, перемещаться между строчками, удалять символы и вставлять новые символы без необходимости создания специального алгоритма.

При запуске приложения, пользователь видит пустую область для редактирования текста, представленную компонентом TMemo (Memo1).

Пользователь может поместить курсор в область TMemo и начать вводить текст с клавиатуры. При этом текст автоматически отображается и обновляется в области TMemo.

В случае если пользователь хочет выполнить некоторые редактирования текста, они могут выделить текст и выполнить операции (например, копировать, вставлять или вырезать). Все эти действия связаны с обработчиками событий для соответствующих элементов меню, таких как MenuItem9Click (вырезать), MenuItem10Click (копировать) и MenuItem11Click (вставить).

Алгоритм сохранения существующего файла и сохранение файла с выбором нового имени и местоположения (Рисунок 10).



Рисунок 10 – Алгоритм сохранения файла.

Аналогичным образом, пользователь может сохранить текст, набранный в компоненте TMemo, с помощью MenuItem6Click (сохранить файл) и MenuItem7Click (сохранить файл с новым именем, так называемый «сохранить как»).

В данной программе сохранение файла реализовано через две функции: сохранение существующего файла и сохранение файла с выбором нового имени и местоположения (Save As).

Эта процедура выполняет следующие действия:

* Запускает SaveDialog1, который открывает окно для выбора местоположения и имени файла.
* Если пользователь выбирает местоположение и имя файла, затем сохраняет содержимое Memo1 в выбранный пользователем файл.
* Обновляет переменную FileWork, присваивая ей имя сохраненного файла.

Сохранение существующего файла:

* Проверяет, содержит ли переменная FileWork имя файла (если нет, значит, пользователь работает с новым файлом).
* Если переменная FileWork пуста, вызывает процедуру SaveAs, позволяя пользователю выбрать имя и местоположение файла.
* Если переменная FileWork содержит имя файла (существует файл), сохраняет содержимое Memo1 в файл, указанный в переменной FileWork.

Алгоритмы двух способов закрытия приложения (Рисунок 11).

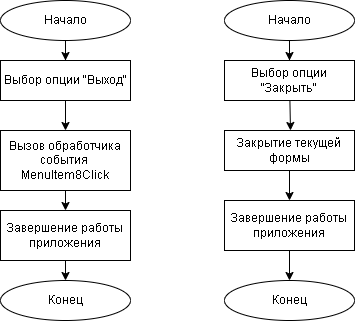


Рисунок 11– Алгоритмы закрытия приложения.

В этом примере кода приложение закрывается, когда пользователь кликает на пункт меню «Выход». Обработка этого события определяется в процедуре TForm1.MenuItem8Click, которая вызывает метод Close.

Здесь Close («Закрыть»)- это метод компонента TForm (в данном случае - TForm1), который закрывает форму. Поскольку это основная форма приложения, когда она закрывается, приложение тоже завершает свою работу.

В этом алгоритме закрытия приложения опцию закрытия инициирует сам пользователь, выбирая пункт меню «Закрыть» или действуя аналогичным образом через другой интерфейс. Указанный обработчик события затем закрывает форму приложения, после чего операционная система освобождает память и другие ресурсы, связанные с приложением. В итоге, приложение полностью закрывается и завершает работу.

## 4.3 Реализация программы. Разработка пользовательского интерфейса

Для реализации программы был выбран язык программирования Object Pascal, среда разработки Lazarus. Для создания пользовательского интерфейса текстового редактора, используется компонент «memo» («текстовый блокнот») из пакета компонентов, предоставляемых Lazarus. Компонент «memo» представляет собой окно, в которое пользователь может вводить и редактировать текст. Исходный код представлен в приложении А.

Реализация программы представлена в разделе implementation, который содержит код для всех обработчиков событий, задекларированных ранее в разделе interface и TForm1.

Unit declaration и интерфейс (interface section):  
В разделе interface объявлены класс TForm1 и несколько компонентов пользовательского интерфейса, таких как Memo1, MainMenu1, OpenDialog1, SaveDialog1, и т.п. Кроме того, в разделе interface определены обработчики событий (event handlers) для каждого из меню.

**Инициализация компонентов и формы:**

– Загружаются необходимые модули и библиотеки.

– Создается класс TForm1, представляющий главную форму приложения.

– Объявляются компоненты формы, такие как FontDialog1, MainMenu1, Memo1, OpenDialog1, SaveDialog1, и элементы меню.

**Объявление глобальных переменных:**  
В программе объявлены две глобальные переменные - Form1: TForm1 и FileWork: string. Переменная FileWork содержит имя текущего редактируемого файла.

**Реализация метода SaveAs:**  
Этот метод открывает диалоговое окно сохранения файла и сохраняет содержимое Memo1 в выбранный файл. Если файл сохранен успешно, переменная FileWork обновляется и содержит имя файла для дальнейшей работы.

**Обработчики событий:**

* MenuItem4Click: Устанавливает FileWork в пустую строку и очищает Memo1, создавая новый пустой файл.
* MenuItem5Click: Открывает диалог OpenFileDialog, отображающий окно для открытия файла. После выбора файла, имеющийся текст загружается в Memo1, после чего переменная FileWork обновляется.
* MenuItem6Click: Сохраняет содержимое Memo1 в текущем файле или вызывает метод SaveAs, если FileWork не содержит имени файла.
* MenuItem7Click: Вызывает метод SaveAs, отображая диалог для сохранения файла.
* MenuItem8Click: Закрывает приложение.
* MenuItem9Click: Вырезает текст из Memo1 и копирует его в буфер обмена.
* MenuItem10Click: Копирует текст из Memo1 в буфер обмена.
* MenuItem11Click: Вставляет текст из буфера обмена в Memo1.
* MenuItem12Click: Включает или отключает перенос слов в Memo1 и изменяет настройки полос прокрутки в соответствии с активированным переносом слов.
* MenuItem13Click: Открывает диалоговое окно FontDialog для выбора шрифта и применяет выбранный шрифт к Memo1.

**Реализация implementation:**  
Раздел implementation содержит реализацию всех идентификаторов и объявленных методов, исполняемых в программа, и объединяет все вышеперечисленные элементы в единую целостность.

**В программном коде реализуются следующие функции:**

**MenuItem4Click** (Создание нового файла). При вызове этой функции делаются следующие действия:

* Устанавливается переменная FileWork в значение пустой строки (''). Это означает, что текущее имя файла сбрасывается, и при сохранении потребуется выбрать новое имя.
* Вызывается метод Memo1.Clear для очистки текстовой области (TMemo). Это удаляет все текущий текст и создает пустую область для написания нового текста.

Исходный код функции представлен на рисунке 12.

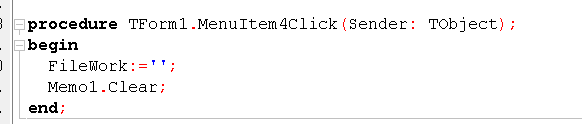


Рисунок 12 – Исходный код функции MenuItem4Click.

**MenuItem5Click** (Открытие существующего файла). Когда пользователь выбирает «Open» (MenuItem5) в меню, происходит следующее:

* Вызывается обработчик событий для MenuItem5, «MenuItem5Click».
* OpenDialog1.Execute запускает диалоговое окно открытия файла. Если пользователь успешно выбирает файл и нажимает кнопку «Открыть», диалог возвращает значение True.
* Если диалог вернул True, Memo1.Lines.LoadFromFile(OpenDialog1.FileName) загружает содержимое выбранного файла в компонент Memo, с которым взаимодействует пользователь.
* Значение FileWork обновляется, чтобы сохранить полное имя выбранного файла. Эта переменная используется для проверки того, открыт ли уже файл при выполнении операций сохранения.

Исходный код представлен на рисунке 13.

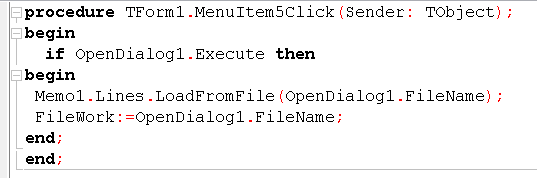


Рисунок 13 – Исходный код функции **MenuItem5Click.**

**MenuItem9Click** (Вырезать выбранный текст). Обработчик событий для пункта меню Вырезать (MenuItem9) вызывается при выборе этой функции в меню «Edit»:

* Событие «MenuItem9Click» активируется при выборе пункта меню Вырезать.
* Memo1.CutToClipboard вызывает метод компонента Memo, которая копирует выделенный текст в буфер обмена и удаляет его из Memo.

Исходный код представлен на рисунке 14.

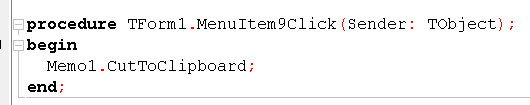


Рисунок 14 – Исходный код функции MenuItem9Click.

**MenuItem10Click (**Копировать выбранный текст). Обработчик событий для пункта меню Копировать (MenuItem6) вызывается при выборе этой функции в меню «Edit»:

* Событие «MenuItem6Click» активируется при выборе пункта меню Копировать.
* Memo1.CopyToClipboard вызывает метод компонента Memo, который копирует выделенный текст в буфер обмена.

Таким образом, пользователь может выбрать текст, используя функцию «Копировать», и затем вставить его в другом месте программы или другом приложении, используя буфер обмена. Отличие функции «Копировать» от функции «Вырезать» в том, что функция «Копировать» не удаляет текст из компонента Memo.

Исходный код представлен на рисунке 15.

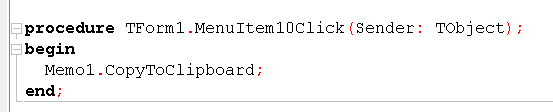
****

Рисунок 15 – Исходный код функции MenuItem10Click.

MenuItem11Click (Вставить текст из буфера обмена). Обработчик событий для пункта меню «Вставить» (MenuItem7) вызывается при выборе этой функции в меню «Edit»:

* Событие «MenuItem7Click» активируется при выборе пункта меню «Вставить».
* Memo1.PasteFromClipboard вызывает метод компонента Memo, который вставляет текст из буфера обмена в текущую позицию курсора в окне редактирования.

Исходный код представлен на рисунке 16.

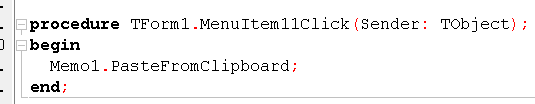
****

Рисунок 16 – Исходный код функции MenuItem11Click.

**MenuItem13Click** (Изменение шрифта). Обработчик событий для пункта меню выбора шрифта (FontItem) вызывается при выборе этой функции в меню «Format»:

* Событие «FontItemClick» активируется при выборе пункта меню выбора шрифта.
* if FontDialog1.Execute then проверяет, был ли выбран новый шрифт пользователем через диалоговое окно Font Dialog.
* Если пользователь выбрал новый шрифт, то Memo1.Font := FontDialog1.Font устанавливает выбранный шрифт для компонента Memo.

Таким образом, пользователь может выбрать новый шрифт и его свойства для выделенного текста, используя функцию «Выбрать шрифт». Когда пользователь выбирает эту функцию, диалоговое окно Font Dialog открывается, где пользователь может выбрать новый шрифт и его свойства. После выбора нового шрифта, он должен быть установлен для выделенного текста в окне редактирования.

Исходный код представлен на рисунке 17.

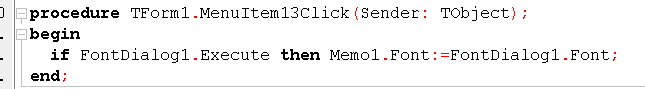


Рисунок 17 – Исходный код функции MenuItem13Click.

**MenuItem12Click** (Переключение режима переноса строк). Обработчик событий для пункта меню переноса строк (WordWrapItem) вызывается при выборе этой функции в меню «Format»:

* Событие «WordWrapItemClick» активируется при выборе пункта меню переноса строк.
* if Memo1.WordWrap then проверяет, включен ли режим переноса строк в компоненте Memo.
* Если режим переноса строк включен, то Memo1.WordWrap := False отключает его.
* Если режим переноса строк выключен, то Memo1.WordWrap := True включает его.

Таким образом, пользователь может выбрать, переносить ли текст на новую строку автоматически, когда он достигнет правой границы окна редактирования, используя функцию «Перенос строк». Если этот режим включен, то каждая строка текста будет автоматически переноситься на новую строку при достижении правой границы окна редактирования. Если этот режим выключен, то каждая строка текста будет продолжаться за пределами правой границы окна редактирования и пользователь должен будет использовать горизонтальную полосу прокрутки, чтобы просмотреть строку.

Исходный код представлен на рисунке 18.

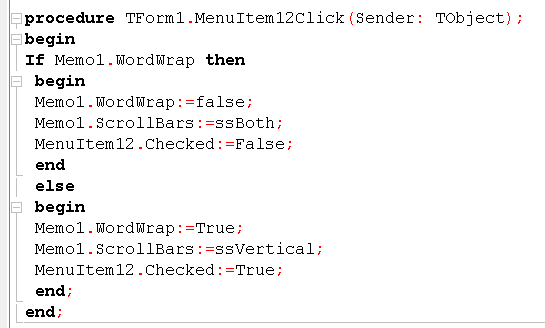
****

Рисунок 18 – Исходный код функции MenuItem12Click.

**MenuItem8Click** (Закрытие приложения). Обработчик событий для пункта меню закрыть (MenuItem8) вызывается при выборе этой функции в меню «File»:

* Событие «MenuItem8Click» активируется при выборе пункта меню Закрыть.
* Close вызывает метод формы, чтобы закрыть текущее окно, которое является экземпляром TForm1.

Таким образом, при выборе функции «Закрыть» пользователь закроет текущее окно программы. Если приложение содержит несколько окон, каждое окно может иметь свою собственную функцию «Закрыть» для закрытия своего окна.

Исходный код представлен на рисунке 19.

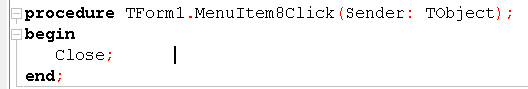


Рисунок 19 – Исходный код функции MenuItem8Click.

**MenuItem6Click** (Сохранение файла) и **MenuItem7Click** (Сохранить файл под другим именем или в другом месте). Обработчик событий для пункта меню сохранения файла (SaveItem) вызывается при выборе этой функции в меню «File»:

* Событие «SaveItemClick» активируется при выборе пункта меню сохранения файла.
* if (SaveDialog1.Execute) then проверяет, был ли выбран путь для сохранения файла через диалоговое окно Save Dialog.
* Если пользователь выбрал путь для сохранения файла, то Memo1.Lines.SaveToFile(SaveDialog1.FileName) сохраняет текущий документ в выбранном файле.
* Caption := 'Text Editor - ' + SaveDialog1.FileName устанавливает новый заголовок окна редактирования, включающий имя сохраненного файла.

Таким образом, пользователь может сохранить активный документ, используя функцию «Сохранить». Когда пользователь выбирает эту функцию, диалоговое окно Save Dialog открывается, где пользователь может выбрать путь для сохранения файла. После выбора пути, документ сохраняется в выбранном файле, и новый заголовок окна редактирования отображает имя сохраненного файла.

Исходный код представлен на рисунке 20.

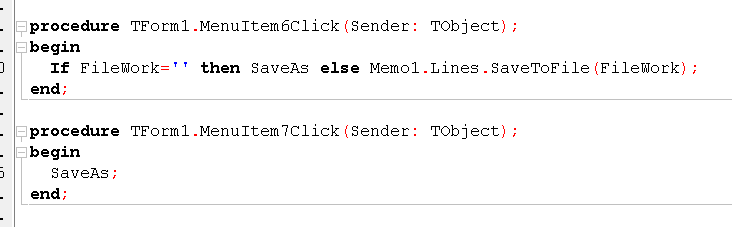


Рисунок 20 – Исходный код функции MenuItem8Click.

**Разработка пользовательского интерфейса.**

На рисунке 21 представлен интерфейс приложения «Блокнот».

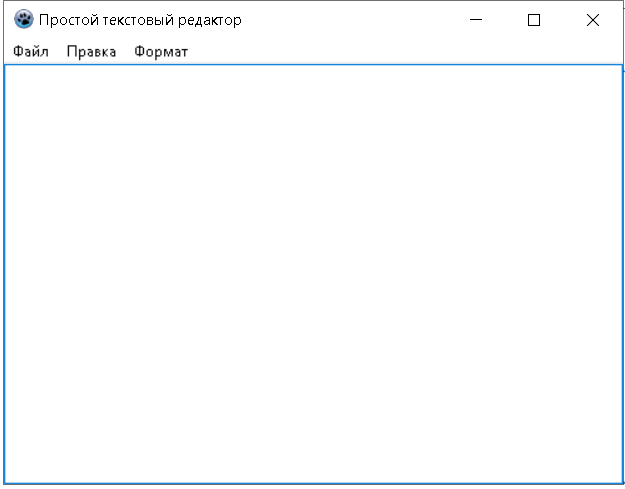


Рисунок 21 – Интерфейс приложения. Главная форма.

Интерфейс программы создается с использованием компонентов и элементов управления

Программа начинается с определения формы TForm1, которая является основным окном программы. Форма содержит различные элементы управления, такие как кнопки, текстовые поля, меню и т. д.

Внутри формы TForm1 находятся объявления компонентов, таких как TFontDialog, TMainMenu, TMemo, TOpenDialog, и TSaveDialog. Эти компоненты представляют собой элементы интерфейса и используются для выполнения различных задач, таких как выбор шрифта, работа с файлами и текстовыми данными и другие.

Компоненты настраиваются с использованием свойств и методов. Например, компонент Memo1 для работы с текстом, который настроен с помощью различных методов, таких как Lines.LoadFromFile и Lines.SaveToFile, чтобы загружать и сохранять текст из/в файл.

Также код включает обработчики событий для различных действий, таких как клики на элементах меню. Например, MenuItem10Click обрабатывает событие клика на определенном элементе меню.

Запуск приложения: Приложение запускается с помощью точки входа. Форма TForm1 становится основным окном приложения.

Обновление и взаимодействие: Приложение может обновлять интерфейс, реагировать на действия пользователя и выполнять различные задачи.

Описание элементов на Главном окне:

* Кнопка «Создать»: Очищает содержимое текстового редактора (Memo1) и устанавливает значение переменной обрабатываемого файла (FileWork) в пустую строку, указывая, что в данный момент нет рабочего файла.
* Кнопка «Открыть»: Открывает диалоговое окно для выбора текстового файла, который необходимо открыть. После выбора файла его содержимое загружается в Memo1, а FileWork устанавливается в имя выбранного файла.
* Кнопка «Сохранить»: Проверяет, есть ли текущий рабочий файл. Если есть, то сохраняет содержимое Memo1 в этот файл. Если нет, вызывает процедуру SaveAs для сохранения файла под новым именем.
* Кнопка Сохранить как: Вызывает процедуру SaveAs для сохранения содержимого Memo1 под новым именем файла.
* Кнопка «Выход»: Закрывает приложение текстового редактора.
* Кнопка «Вырезать»: Вырезает выделенный текст в Memo1 в буфер обмена.
* Кнопка «Копировать»: Копирует выделенный текст в Memo1 в буфер обмена.
* Кнопка «Вставить»: Вставляет содержимое буфера обмена в Memo1.
* Кнопка «Перенос слов»: Переключает режим переноса слов в Memo1 и соответствующим образом настраивает полосы прокрутки.
* Кнопка «Шрифт»: Открывает диалоговое окно (Рисунок 22) шрифта для выбора шрифта пользователем. После выбора шрифта содержимое Memo1 отображается выбранным шрифтом.

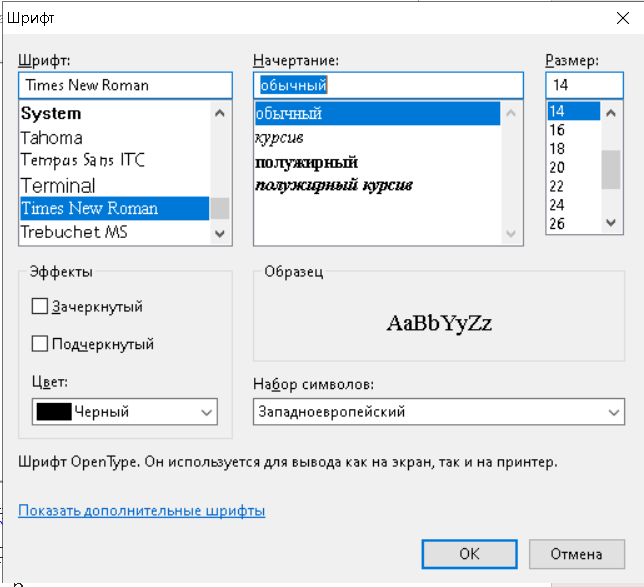


Рисунок 22 – Диалоговое окно шрифта.

Диалоговое окно выбора шрифта предоставляет пользователю возможность изменить шрифт, его стиль и размер в приложениях. В предоставленном коде используется компонент TFontDialog для реализации данной функциональности.

## 4.4 Тестирование программы.

Тестирование программы проведено в соответствии с Программой и методикой исследования, написанной в процессе разработки программного продукта.

Таблица 1 – Результаты испытаний

| № | Шаг испытаний (проверок) | № пункта  Методики | Отметка о прохождении (да/нет) | Примечания |
| --- | --- | --- | --- | --- |
| 1 | Проверка соответствия технического задания требованиям, предъявляемым к студентам направления подготовки по профильным дисциплинам. | 1 | Да |  |
| 2 | Проверка соответствия программы и методики тестирования требованиям, предъявляемым к студентам направления подготовки по профильным дисциплинам. | 1 | Да |  |
| 3 | Проверка соответствия руководства пользователя требованиям, предъявляемым к студентам направления подготовки по профильным дисциплинам. | 1 | Да |  |
| 6 | Проверка на корректность создания формы . | 2 | Да |  |
| 7 | Проверка на корректность активации кнопок. | 2 | Да |  |

Таблица 1 – Продолжение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 8 | Проверка функции кнопки «Создание» на создание нового файла | 2 | Да |  |
| 9 | Проверка функции кнопки «Открыть» на открытие диалогового окна для выбора файла. | 3 | Да |  |
| 10 | Проверка функции кнопки «Сохранить» на сохранение содержимого формы в файл. | 4 | Да |  |
| 12 | Проверка функции кнопки «Сохранить как» на сохранение содержимого формы под новым именем файла. | 5 | Да |  |
| 13 | Проверка функции кнопки «Выход» на закрытие приложения. | 5 | Да |  |
| 14 | Проверка функции кнопки «Перенос по словам» на переключение в режим переноса слов. | 7 | Да |  |
| 15 | Проверка функции кнопки «Вырезать» на вырезание выделенного текста в буфер обмена. | 8 | Да |  |
| 16 | Проверка функции кнопки «Копировать» на копирование выделенного текста в буфер обмена. | 8 | Да |  |

Таблица 1 – Продолжение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 17 | Проверка функции кнопки «Вставить» на вставление содержимое буфера обмена. | 8 | Да |  |
| 18 | Проверка функции кнопки «Шрифт» на открытие диалогового окна для выбора шрифта. | 9 | Да |  |
| 19 | Проверка формата сохранения формы. | 10 | Да |  |
| 21 | Проверка места сохранения формы. | 10 | Да |  |
| 22 | Проверка скорости генерации формы. | 11 | Да |  |
| 23 | Проверка скорости сохранения формы. | 11 | Да |  |
| 24 | Проверка эффективности изменения параметров настройки генерации. | 11 | Да |  |
| 25 | Проверка на возникновение системных ошибок. | 12 | Да |  |
| 26 | Проверка корректной работы игры при попытках вызвать системную ошибку. | 12 | Да |  |

# ЗАКЛЮЧЕНИЕ

Для создания пакета документации были изучены ГОСТы и требования к написанию технического задания, программы и методики тестирования, руководству пользователя. Благодаря этому я получила навыки по написанию документов.

В рамках индивидуального задания было разработано приложение «Блокнот», в ходе которого были изучены и повторены многие компоненты, функции языка программирования Lazarus IDE. Также изучены различные сайты по программированию на данном языке, благодаря этому появились навыки в правильном и грамотном создании интерфейса и написании кода приложения.

Таким образом, программа учебной практики ПМ.05 в период с 09.01.2023 по 14.06.2023 была выполнена мной в полном объеме.

При выполнении работ, предусмотренных заданием на практику, в организации ФГБОУ ВО «Вятский государственный университет» мной, Щекотовой Екатериной Алексеевной, соблюдались правила внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологические правила и гигиенические нормативы. Пройден вводный инструктаж и инструктаж на рабочем месте. Замечаний не имела.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Программирование на Lazarus [Электронный ресурс]. – <http://intuit.valrkl.ru/course-1265/>
2. Основы алгоритмизации и программирования: учебник / И. Г. Семакин, А. П. Шестаков. - Москва: Академия, 2018. - 300 с.
3. Lazarus – бесплатная среда по разработке программного обеспечения [Электронный ресурс]. – <https://lazarus-rus.ru/>
4. Lazarus (программное обеспечение) – Lazarus (software) [Электронный ресурс]. – <https://ru.xcv.wiki/wiki/Lazarus_(IDE)>
5. Lazarus [Электронный ресурс]. – <https://spravochnick.ru/informatika/lazarus/>

# ПРИЛОЖЕНИЕ А

**ИСХОДНЫЙ КОД**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, Menus,

StdCtrls;

type

{ TForm1 }

TForm1 = class(TForm)

FontDialog1: TFontDialog;

MainMenu1: TMainMenu;

Memo1: TMemo;

MenuItem1: TMenuItem;

MenuItem10: TMenuItem;

MenuItem11: TMenuItem;

MenuItem12: TMenuItem;

MenuItem13: TMenuItem;

MenuItem2: TMenuItem;

MenuItem3: TMenuItem;

MenuItem4: TMenuItem;

MenuItem5: TMenuItem;

MenuItem6: TMenuItem;

MenuItem7: TMenuItem;

MenuItem8: TMenuItem;

MenuItem9: TMenuItem;

OpenDialog1: TOpenDialog;

SaveDialog1: TSaveDialog;

procedure MenuItem10Click(Sender: TObject);

procedure MenuItem11Click(Sender: TObject);

procedure MenuItem12Click(Sender: TObject);

procedure MenuItem13Click(Sender: TObject);

procedure MenuItem4Click(Sender: TObject);

procedure MenuItem5Click(Sender: TObject);

procedure MenuItem6Click(Sender: TObject);

procedure MenuItem7Click(Sender: TObject);

procedure MenuItem8Click(Sender: TObject);

procedure MenuItem9Click(Sender: TObject);

private

{ private declarations }

public

{ public declarations }

end;

var

Form1: TForm1;

FileWork: String;

implementation

{$R \*.lfm}

Procedure SaveAs;

begin

If Form1.SaveDialog1.Execute then

begin

Form1.Memo1.Lines.SaveToFile(Form1.SaveDialog1.FileName);

FileWork:=Form1.SaveDialog1.FileName;

end;

end;

{ TForm1 }

procedure TForm1.MenuItem9Click(Sender: TObject);

begin

Memo1.CutToClipboard;

end;

procedure TForm1.MenuItem8Click(Sender: TObject);

begin

Close;

end;

procedure TForm1.MenuItem5Click(Sender: TObject);

begin

if OpenDialog1.Execute then

begin

Memo1.Lines.LoadFromFile(OpenDialog1.FileName);

FileWork:=OpenDialog1.FileName;

end;

end;

procedure TForm1.MenuItem6Click(Sender: TObject);

begin

If FileWork='' then SaveAs else Memo1.Lines.SaveToFile(FileWork);

end;

procedure TForm1.MenuItem7Click(Sender: TObject);

begin

SaveAs;

end;

procedure TForm1.MenuItem4Click(Sender: TObject);

begin

FileWork:='';

Memo1.Clear;

end;

procedure TForm1.MenuItem10Click(Sender: TObject);

begin

Memo1.CopyToClipboard;

end;

procedure TForm1.MenuItem11Click(Sender: TObject);

begin

Memo1.PasteFromClipboard;

end;

procedure TForm1.MenuItem12Click(Sender: TObject);

begin

If Memo1.WordWrap then

begin

Memo1.WordWrap:=false;

Memo1.ScrollBars:=ssBoth;

MenuItem12.Checked:=False;

end

else

begin

Memo1.WordWrap:=True;

Memo1.ScrollBars:=ssVertical;

MenuItem12.Checked:=True;

end;

end;

procedure TForm1.MenuItem13Click(Sender: TObject);

begin

if FontDialog1.Execute then Memo1.Font:=FontDialog1.Font;

end;

end.

# ПРИЛОЖЕНИЕ Б

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

# ПРИЛОЖЕНИЕ В

**ПРОГРАММА И МЕТОДИКА ИССЛЕДОВАНИЯ**

# ПРИЛОЖЕНИЕ Г

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**