**

Колледж космического машиностроения и технологий

**КУРСОВОЙ ПРОЕКТ**

**По МДК.01.02 «Прикладное программирование»**

**Тема: «Блокнот»**

Выполнил студент

Головатюк М.В.

Группа П2-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_ (Дата сдачи работы)

Королев, 2020

**Оглавление**

[Введение 4](#_Toc24656041)

[1. Теоретическая часть 5](#_Toc24656042)

[1.1. Что такое блокнот 5](#_Toc24656043)

[2. Проектная часть](#_Toc24656050) 16

[2.2. Описание ключевых функций 16](#_Toc24656054)

[3. Организационная часть 18](#_Toc24656057)

[3.1. Порядок установки программного средства 18](#_Toc24656058)

[3.2. Руководство оператора 18](#_Toc24656059)

[3.3. Минимальные условия работы программы 18](#_Toc24656060)

[Заключение 22](#_Toc24656061)

[Список источников](#_Toc24656062) 22

[Приложение 1 23](#_Toc24656063)

# Введение

Разработка проекта будет вестись на языке «Java» с использованием среды разработки программного обеспечения «IntelliJ IDEA Community Edition 2019». Данная программа должна будет содержать в себе редактор текста возможность открытия и сохранения файлов, для этого необходимо создать удобный интерфейс программы.

В теоретической части будет рассмотрено введение в предметную область: будет определено понятие «Блокнот», его устройство и функционирование.

В проектной части будет приведено описание использовавшихся при разработке проекта программного инструмента, описание его интерфейса и функционала, а также рассмотрены листинги (исходный код) ключевых программных модулей проекта.

В организационной части будут описаны минимальные требования для работы программы, её назначение, порядок установки и описание интерфейса.

В заключении будут сделаны общие выводы о проекте, определены его возможные пути доработки и возможные области применения.

Для осуществления обозначенных целей служат следующие задачи:

* 1. Описание инструмента разработки проекта.
  2. Разработка программы и её функции.
  3. Написание кода программы.
  4. Описание руководства пользователя.

# Теоретическая часть

## Что такое блокнот

1. **Блокнот Windows** - простой текстовый редактор, являющийся частью операционных систем Microsoft Windows, начиная с вышедшей в 1985 году Windows 1.0.

Блокнот использует оконный класс *EDIT*. Вплоть до вышедшей в 2000 году Windows ME поддерживались только самые базовые функции, многие функции были доступны только из меню, а максимальный размер файла составлял 64 килобайта (предел класса EDIT). В настоящее время редактор поддерживает контекстную замену, горячие клавиши (например, Ctrl-S для сохранения файла), снят предел в 64 Кбайт и добавлена поддержка Юникода.

Кроме Windows, Блокнот способен выполняться также в ReactOS и Wine.

Альтернативой Блокноту является текстовый редактор MS-DOS (EDIT.COM), который можно вызвать из командной строки в виде «edit».

Блокнот до Windows 10 не способен корректно работать с файлами в текстовом формате Unix, где символом переноса является байт с кодом 10, в отличие от Windows и DOS, где используются байты 13, 10.

2. **Bred** — бесплатный текстовый редактор для Windows с закрытым исходным кодом и сравнительно небольшим набором функций. Позиционируется как более совершенная и удобная замена стандартным редакторам Блокнот и WordPad, входящим в состав операционных систем семейства Microsoft Windows.

Текстовый редактор был создан небольшой ульяновской компанией Gladiators Software. Ранние версии редактора из линейки 2.x.x (Bred-2), вплоть до последней 2.1.8, выпущенной 24 декабря 2000 года, предназначались для работы в Windows 9x/NT4 и были написаны на чистом Windows API в среде Delphi (Object Pascal), поэтому были очень компактными по объёму и быстрыми, чем, вкупе с удобством использования и достаточно богатым, по сравнению со стандартным Блокнотом Windows, набором функций, заслужили популярность среди пользователей.

Список поддерживаемых функций редактора Bred-2 включал в себя среди прочего: все функции стандартного Блокнота Windows, поддержку ANSI, KOI8 и OEM кодировок, с автоматическим их распознаванием и возможностью конвертирования из одной кодировки в другую, а также обычного текста в транслитерированный, выравнивание текста (по краям, центру и ширине), отсутствие ограничений (кроме объёма свободной оперативной памяти) по откату/возврату изменений и объёму текста (стандартный Блокнот в Windows 9x/NT4 позволял создавать и открывать файлы объёмом не более 64 килобайт), запоминание позиции курсора при закрытии документа, подсветка HTML-кода, перетаскивание текста мышкой и другие возможности.

При условии наличия в каталоге с программой библиотеки riched32.dll из дистрибутива Windows NT4, она способна работать и в более поздних операционных системах, например, Windows 2000, ME, XP и даже Windows 7. Ввиду своей крайней в прошлом популярности эта версия до сих пор доступна на официальном сайте. Распространяется в двух редакциях: с русским и английским интерфейсом.

Разработка следующего поколения редактора, предназначенного для работы в современных операционных системах, в связи с различными затруднениями и сменой среды разработки с Delphi на C++ растянулась на несколько лет. Первая публичная бета-версия обновленного редактора под названием Bred-3 вышла в свет 7 июня 2004 года. В ней был полностью переписан текстовый движок, удалены некоторые старые возможности и добавлены новые, в том числе полная поддержка Windows 2000/XP/2003 (для Windows 98/ME также существует отдельный дистрибутив), поддержка Юникод (кодировки UTF-8/UTF-16), подсветка синтаксиса языков программирования (через библиотеку Colorer), поддержка плагинов (кроме Colorer, входящего в комплект программы, больше плагинов выпущено не было), двухоконный режим, возможность автоматического создания резервных копий во время редактирования и другие функции.

**3. Notepad Plus Plus** - предназначен для работы с текстом с огромным количеством функций, которые помогут вам не запутаться в кодах и написать документ, программу или еще либо-что, так как вам этого хочется.

В Notepad Rus удобной покажется функция замены спецсимволов на их коды. Часто такие символы не заметны в огромном количестве знаков на языках программирования, их можно заменить на коды, использовав, только эту функцию и не перебирая весь текст вновь и вновь при помощи XML Notepad.

**Основные возможности Notepad++:**

 — Подсветка синтаксиса для различных языков программирования: C, C++, HTML, ASP, Java, Pascal, CSS и многих других;

— Функция работы одновременно с несколькими документами;

— Авто-завершение набираемого слова.

 — Возможность создания собственного списка API функций (или скачать его со страницы загрузки).

 — Поддержка регулярных выражений Поиска/Замены.

 — Полная поддержка перетягивания фрагментов текста.

 — Динамическое изменение окон просмотра.

 — Автоматическое определение состояния файла (уведомление об изменении или удалении файла другой программой — с возможностью перезагрузить файл или удалить его из программы).

 — Увеличение и уменьшение (масштабирование).

 — Поддержка большого количества языков.

 — Листинг программ можно печатать в цвете — так, как вы видите на экране в окне редактирования.

 — Заметки о документе.

 — Выделение скобок при редактировании текста.

 — Запись макроса и его выполнение.

4. **Akelpad** - **Что такое AkelPad и зачем нужна эта программа?** Не секрет, что значительная часть функций текстовых процессоров (таких, как Word) не востребована большинством их пользователей, а на производительность и скорость эти функции влияют достаточно сильно. Сейчас множество людей пользуется ноутбуками и другими портативными электронными устройствами, значительная часть которых достаточно ограничена в своих возможностях, поэтому производительность очень важна.

Так что нередко стоит предпочесть текстовые редакторы, а не текстовые процессоры. Самым популярным из них является всем известный «Блокнот», который входит в стандартный набор Windows и имеет достаточно неплохой набор функций.

Но вот всё-таки возможностей «Блокнота» может оказаться уже маловато! В таком случае я нашёл для себя отличный вариант — AkelPad, он очень похож на Notapad++!

**Программа AkelPad**— это свободный текстовый редактор, имеющий открытый исходный код.

**Возможности AkelPad**

Изначально программа обладает минимумом необходимых функций, но их число можно расширить за счёт плагинов.

С версии 4.0.0 alpha 1 в AkelPad применяется собственный контрол под названием «AkelEdit» (вместо Rich Edit от MS Windows), что даёт новые возможности по управлению текстом и его поведением в окне.

**Итак, набор возможностей приложения AkelPad:**

Режимы — однооконный (SDI), многооконный (MDI) и псевдо-многооконный (PMDI);

Полная поддержка на Unicode системах Unicode строк;

Работа с различными кодировками Unicode, любой установленной в системе кодовой страницей;

Работа при переводе строки с Unic, DOS/Windows, Mac форматами;

Предпросмотр файлов до их открытия. Это очень удобно, когда среди многих файлов, имеющих одинаковые названия, нужно найти один, который и имеет нужное содержание;

Известно, что это у программы AkelPad есть возможность корректно отображать любые псевдографики;

Возможность выделять текст блоками;

Многоуровневый откат любых действий;

Быстрые замена/поиск строк в тексте;

Запоминание в файле кодировки, позиции курсора;

Печать (в том числе и цветная), предпросмотр;

В AkelPad есть поддержка различных языковых модулей.

5. **SkimEdit** - Первое отличие от Блокнота заключается в возможности редактирования документов более 64КБ. Ограничение поднимается до 2 ГБ. Некоторые редакторы и вовсе объявляют о возможности работы с файлами неограниченного объема. Но и два гигабайта тоже, мягко говоря, весьма солидный объем для простого текста.

Программа поддерживает все основные кодировки, использующиеся при работе с простым текстом. Мало того, вы можете свободно преобразовывать кодовые страницы. Некоторые из основных сочетаний вынесены в отдельные пункты меню, остальное собрано воедино. Вы указываете исходную кодировку и ее конечный вариант, после чего происходит преобразование текста.

Некоторые текстовые редакторы обладают возможностью подсветки синтаксиса. Можно сказать, что и SkimEdit умеет делать подобное. Но нужно сделать одну существенную оговорку. В отличие от других программ, где набор вариантов синтаксиса огромен, здесь вы можете выбирать лишь два режима. Один из них выделяет URL, и вы можете даже переходить по ссылкам, автоматически запуская браузер, установленный в системе по умолчанию. Второй режим заключается в подсветке синтаксиса HTML.

Качество программного продукта складывается не только из серьезных факторов, но и из мелочей. Текстовые редакторы часто используются для быстрого просмотра документов. Вам хочется открыть файл, быстро закрыть и перейти на следующий элемент списка. Все программы закрываются с помощью сочетания клавиш Alt+F4. Это неудобно. SkimEdit закрывается также с помощью Esc.

Вторая приятная мелочь заключается в автоматическом копировании информации в буфер обмена. Подобный механизм активно используется в открытых системах, но в Windows применяется довольно редко. Для копирования выделенного текста в буфер, по штату необходимо нажать Ctrl+C. В случая использования SkimEdit, копирование происходит автоматически после завершения выделения. Для полного счастья приложению не хватает возможности вставки текста с помощью средней кнопки мыши.

Текстовый редактор позволяет выделять и удалять строки, стирать все переводы строк, тем самым удаляя абзацы. Вызов инструмента поиска осуществляется с помощью одной клавиши, а не какого-либо их сочетания. Программа умеет сохранять до ста имен открытых файлов с записью последнего положения курсора.

6. **TEA** - Программа имеет только русский интерфейс, что уже чуть ли не нонсенс. Популярные приложения, кто бы ни был их автором, всегда имеют, в первую очередь, английский язык в меню, в диалоговых окнах. Остальные языки добавляются опционально. Здесь же мы имеем совершенно иную картину, которая, впрочем, ничуть не портит общее впечатление от продукта.

Программа имеет две версии. Одна из них работает в Windows, а другая - в Linux. Они не идентичны, обладая разным набором инструментов. И это, в общем-то, тоже редкость.

TEA не имеет панели инструментов. Сразу под главным меню располагается строка с вкладками. Внутри нее отображаются заголовки открытых документов, между которыми происходит переключение. Под областью редактирования текста располагается панель вывода служебной информации. В качестве нее выступают результаты работы некоторых функций, сообщения об ошибках или удачно выполненных действиях. Сразу над строкой статуса расположено поле ввода. Именно так оно и называется. Поле используется для ввода различных числовых и строковых значений, параметров используемых функций. Например, с помощью поля ввода осуществляется поиск и замена текста.

Внутри заголовка рабочего окна приложения присутствует кнопка с изображением буквы Z. Данный элемент интерфейса предназначается для особого сворачивания TEA. На экране остается лишь заголовок окна и главное меню. Повторное нажатие кнопки Z возвращает окну развернутый вид.

Нижняя часть рабочего окна отводится под панель состояния. На ней отображается формат окончания строк (Windows или UNIX), текущая кодировка текста и раскладка клавиатуры.

При перетаскивании объектов на рабочее окно TEA, может произойти несколько вариантов действий. Если осуществляется перенос текстовых файлов, то происходит их открытие. Если же ведется работа с графическими файлами, то изображения вставятся в текст в виде элементов того языка разметки, который выбран в настройках приложения.

TEA поддерживает шаблоны. Они хранятся внутри специальной папки. Любой текстовый документ можно сохранить как шаблон.

Программа хранит историю открытых файлов, а также позволяет вести список избранных документов. В нем сохраняются не только ссылки, но и указания на кодировку текста.

Во многих программах поиск текста осуществляется путем вызова специального диалогового окна. Реже встречается поиск по мере набора текста. В TEA используется третий вариант. Внутри специального поля, расположенного в нижней части рабочего окна приложения, вводится текст поискового запроса. После этого вы выбираете функцию, которую необходимо применить к данной строке. В их число входит поиск и различные варианты замены.

Отдельного внимания заслуживает замена по таблице. Она применяется в тех случаях, когда необходимо сразу заменить несколько слов на другие и, возможно, требуется повторение процедуры в нескольких документах. Таблица состоит из набора равенств. В их левой стороне указываются исходные строки, а справа вписываются конечные аналоги.

Текстовый редактор удобно использовать для верстки документов HTML, Docbook и LaTeX. При выделении фрагмента текста и выбора какого-либо пункта меню верстки, регион оформляется тегами. Некоторые теги вставляются непосредственно в текст, не требуя предварительного выделения.

Программа умеет вычислять конечный объем документа в формате HTML. При этом подсчитывается, какой размер имеет текущий документ вместе с картинками, flash-анимацией и прочими объектами, на которые имеются локальные ссылки. Объем измеряется в килобайтах.

TEA позволяет вставлять в документ стандартные строки, например, дату, имя файла, текущий цвет и многое другое. Поле ввода может использоваться в качестве калькулятора. В нем осуществляются различные вычисления. Большое количество функций связано с обработкой строк. Вы можете удалять строки, устранять лишние пробелы, нумеровать их, менять порядок. Символы внутри конкретной строки можно пускать вспять, меняя порядок.

Программа поддерживает создание пользовательских сокращений. Некоторые введенные заранее слова, строки, фразы можно впоследствии вызывать из специального меню.

Допускается создание закладок внутри текста. Вы можете создавать до десяти якорей, к которым осуществляется быстрый доступ из меню @. Кроме того, TEA позволяет осуществлять навигацию от открывающей скобки к ее закрывающему элементу.

TEA, в отличие от большинства более простых текстовых редакторов, требует долгого освоения и привыкания. Он немного похож на популярный в открытых системах Emacs, применяющийся, в основном, при подготовке исходных текстов программных продуктов.

**7. EmEditor** - Текстовый редактор обладает логичным, продуманным интерфейсом. Панели инструментов можно не просто настраивать, добавляя или удаляя на них кнопки, а также свободно перетаскивать, как котят за шиворот, мышкой.

Программа позволяет работать с несколькими документами одновременно. Переключение между ними осуществляется с помощью вкладок. Кроме того, вы можете вызывать специальную панель Open Documents, на которой представлен список редактируемых текстов. Вкладки могут группироваться по проектам, над ними возможен целый набор пакетных операций.

Многие текстовые редакторы способны открывать большие тексты. Разработчики EmEditor Pro не поскупились и реализовали поддержку файлов, имеющих объем более четырех гигабайт. В документе может присутствовать десять миллионов строк и столько же столбцов.

Текстовый редактор имеет продвинутую систему подсветки синтаксиса. Большинство приложений позволяет выполнять лишь один вариант раскрашивания, и чаще всего он выбирается вручную. EmEditor Pro не просто обладает способностью распознавания конструкций многих языков программирования, но также умеет осуществлять комплексное выделение. Если внутри текста встречаются фрагменты, соответствующие разным языкам, разметкам или типам сценариев, то каждый из них выделяется собственным цветом. В правой части рабочего окна приложения находится информационная область, внутри которой указывается название языка программирования. Подсветка синтаксиса в EmEditor Pro

Второй метод выделения синтаксиса задействуется с помощью модуля Outline. Он основывается не на цвете, а на группировке конструкций языков программирования. Исходный текст программы состоит из набора функций. Каждая из них имеет свое начало и конец. Outline позволяет сворачивать функции в заголовок. Результирующее дерево отображается внутри правой боковой панели. Выделение конструкций языка программирования в EmEditor Pro

Выделение цветом поддерживается и печатью. Если вы используете цветной принтер, то подсветка синтаксиса будет перенесена на бумагу.

EmEditor Pro обладает мощными инструментами поиска информации. Вы можете осуществлять его внутри текущего файла, а также указывать их список. Кроме того, существует режим поиска во всех открытых документах. Каков бы ни был его тип, вы можете использовать регулярные выражения.

В текстовом редакторе удаляется немало внимания сохранности документов. Работает функция автоматического сохранения файлов. По умолчанию оно осуществляется каждые десять минут. Кроме того, вы можете создавать резервные копии документов. Они помещаются в мусорную корзину. Если копии оказываются ненужными, то о них никто и не вспомнит, так как корзина со временем очищается. А коли случилось так, что необходимы старые варианты документов, то нетрудно покопаться в виртуальном мусоре и выудить нужный файл.

Программа поддерживает запись макросов. Ее особенностью является то, что вы можете запоминать не только нажатия клавиш, но также движения мыши и нажатия ее кнопок.

С помощью простого, но эффективного инструмента осуществляется предварительный просмотр HTML-страниц. Большинство текстовых редакторов используют вызов Internet Explorer или других браузеров. В EmEditor Pro все намного проще. В нижней части рабочего окна приложения появляется дополнительная панель, внутри которой происходит предварительный просмотр HTML-документа.

EmEditor Pro обладает инструментом сравнения файлов. Внутри левой боковой панели отображаются отличия документов. Идентичные фрагменты показываются традиционным цветом (по умолчанию белый цвет). Отличия первого файла выделяются красным цветом, а уникальные строки второго документа обозначаются желтым оттенком.

С помощью модуля Snippets удобно вводить часто повторяющиеся фрагменты текста. Вы можете организовывать древовидную структуру сокращений. Внутри левой боковой панели находится их список. Вы выбираете необходимый элемент, и текст помещается в документ.

**8. GridinSoft Notepad** может быть использован как для редактирования простых текстов, так и в программировании, при разработке HTML-страниц. Программа имеет мощный, гибко настраиваемый интерфейс. Ее внешний вид меняется с помощью стилей. Они не включают в себя замену дизайна кнопок, но влияют на цветовые решения в оформлении рабочего окна приложения.

Рабочее окно украшает множество панелей инструментов, которые, помимо всего прочего, можно перетаскивать мышью. Они могут располагаться не только в верхней части окна, но и по бокам, а также внизу. Настройки панелей включают в себя добавление и удаление кнопок, ввод текста подсказок и многое другое.

Любая панель может быть превращена в независимое окно. Подобным образом по умолчанию представлен список открытых документов, файловый и навигатор. Другие дополнительные инструменты встраиваются как панели.

Программа поддерживает пятьдесят вариантов подсветки кода. Помимо столь внушительного числа способов раскрашивания текста, вы можете также создавать собственные методы. Поддерживается перенос цвета во время печати документов.

Просмотр содержимого файлов возможен не только в виде текста. Вы можете переключаться в режим HEX и править документ по байтам. Однако для просмотра HTML страницы, в любом случае, приходится запускать стороннее приложение – браузер.

Программа очень старается помогать при наборе текста. Во-первых, работает проверка орфографии. В дистрибутив включен только английский словарь. Его русский аналог загружается отдельно. Проверка работает по мере набора текста. Помимо поиска ошибок, текстовый редактор умеет самостоятельно завершать слова. Данная операция очень помогает при вводе различных команд, функций, операторов. Не обязательно вводить все символы, достаточно вызвать подсказку. Вы можете самостоятельно создавать шаблоны, правила автоматической замены текста. Это еще больше упрощает его ввод, а значит и подготовку документов.

Программа поддерживает поиск текста в обоих направлениях, с учетом регистра, от положения курсора, по целым словам, с использованием регулярных выражений. Поиск может осуществляться внутри текущего документа, а также во всех выбранных пользователем файлах. На панели инструментов присутствует строка ввода запроса. С помощью нее осуществляется поиск по мере набора текста. Все вхождения строки сразу же выделяются желтым цветом.

Навигация по документам упрощается, благодаря наличию возможности создания закладок внутри текста. Их может быть десять штук.

Для более удобного просмотра документа реализована нумерация строк в левой части рабочего окна приложения. Сверху можно отображать линейку. Ее шкала отображает число символов в строках. Функции увеличения и уменьшения размера шрифта вынесены на отдельные кнопки.

Программа позволяет импортировать файлы со сложным форматированием RTF. Они преобразуются в простой текст или в HTML, после чего документы можно свободно редактировать. Экспорт возможен в форматы RTF, HTML и LaTex.

Помимо платной версии программного продукта, можно загрузить его облегченный вариант GridinSoft Notepad Lite, в котором отсутствуют некоторые продвинутые функции.

# Проектная часть

## Описание разработанного приложения

Данная программа имеет возможность редактирования текста, сохранения и открытия файлов.

## Описание ключевых функций

### Описание программной логики проекта

В программе содержатся несколько функций, которые осуществляют работу с текстом: написание и редактирование текста, сохранение и открытие текстовых файлов. Рассмотрим некоторые наиболее важные из них.

Функция написания и редактирования текста.

Код этой функции приведён в листинге 2.1

**Листинг 2.1 «код функции «JTextArea»»**

**public** NoteMain() **throws** Exception { *//Создание JFrame* **this**.setSize(500, 500);  
 **this**.setTitle(**"Блокнот"**);  
 setDefaultCloseOperation(***DISPOSE\_ON\_CLOSE***);  
 **this**.**textArea**.setFont(**new** Font(**""**, Font.***PLAIN***, 14));  
 **this**.getContentPane().setLayout(**new** BorderLayout());  
 **this**.getContentPane().add(**textArea**);  
 **jMenuBar**.add(**File**);  
 **jMenuBar**.add(**Help**);  
 setJMenuBar(**jMenuBar**);  
}

Функция сохранения файла.

Данная функция выполняется в случае, если пользователь нажал на кнопку «Сохранить». Код этой функции приведён в листинге 2.2.

**Листинг 2.2 «код функции «Save» »**

**save**.addActionListener(**new** ActionListener() { *//Создание функции save* **public void** actionPerformed (ActionEvent arg0){  
 JFileChooser save = **new** JFileChooser();  
 **int** option = save.showSaveDialog(save);  
 **if** (option == JFileChooser.***APPROVE\_OPTION***) {  
 **try** {  
 BufferedWriter out = **new** BufferedWriter(**new** FileWriter(save.getSelectedFile().getPath()));  
 out.write(**textArea**.getText());  
 out.close();  
 } **catch** (Exception ex) {  
 System.***out***.println(ex.getMessage());  
 }  
 }  
 }  
});

Функция открытия файла

Открывает диалог открытия файла и в случае положительного результата читает содержимое указанного файла в лист записей. Код этой функции приведён в листинге 2.3

**Листинг 2.3 «код функции «Open» »**

**open**.addActionListener(**new** ActionListener() { *//Создание функции open* **public void** actionPerformed (ActionEvent arg0){  
 JFileChooser open = **new** JFileChooser();  
 **int** option = open.showOpenDialog(open);  
 **if** (option == JFileChooser.***APPROVE\_OPTION***) {  
 **theText**.setText(**""**);  
 **try** {  
 Scanner scan = **new** Scanner(**new** FileReader(open.getSelectedFile().getPath()));  
 **while** (scan.hasNext())  
 **textArea**.setText(scan.nextLine() + **"\n"**);  
 } **catch** (Exception ex) {  
 System.***out***.println(ex.getMessage());  
 }  
 }  
 }  
});

Список компонентов и программных модулей проекта

Список используемых компонентов и программных модулей проекта приведён в таблице 2.1.

Таблица 2.1 Список программных модулей проекта

|  |  |
| --- | --- |
| Файл исходного кода | Описание |
| NoteMain.java | создающийся в проекте, использующим JEditorPane по умолчанию. |
| Note.java | Модуль содержит все функции программы. |

# 

# Организационная часть

## Порядок установки программного средства

Программный модуль не требует установки, т.к. все необходимые файлы идут как встроенные модули.

## Руководство оператора

**Назначение программы**

Программа предназначена для работы с текстовыми файлам.

## Минимальные условия работы программы

**Минимальный состав технических средств**

В состав технических средств должен входить персональный компьютер, включающий в себя:

* Процессор: Intel Pentium G3260;
* Оперативная память (RAM): 1024 Мб;
* Жёсткий диск (доступного места на диске): 512 Мб;
* Дополнительное оборудование: монитор, клавиатура, мышь.

**Минимальный состав программных средств**

Операционная система Microsoft Windows 7 или более современная.

**Требования к пользователю**

Для успешной работы с программой пользователю необходимо:

* Получить навыки работы с операционной системой Windows;
* Ознакомиться с данным руководством пользователя.
  1. **Запуск программы,**

После запуска программы откроется окно с графическим оконным интерфейсом, в котором будут пункты меню и поле для ввода текста

(рисунок 3.1).

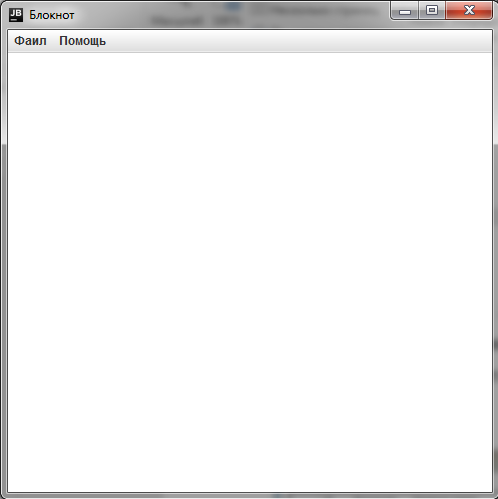


Рисунок 3.1 Окно программы с меню и полем ввода текста.

Имеется возможность сохранить результаты работы с программой. Для этого следует выбрать пункт меню «Файл», затем в выпавшем списке выбрать пункт «Сохранить». На экране появится стандартный диалог сохранения файла, в котором пользователю будет предложено выбрать имя сохраняемого файла и его расположение. При подтверждении файл будет сохранён в выбранной директории с указанным именем (Рисунок 3.2)

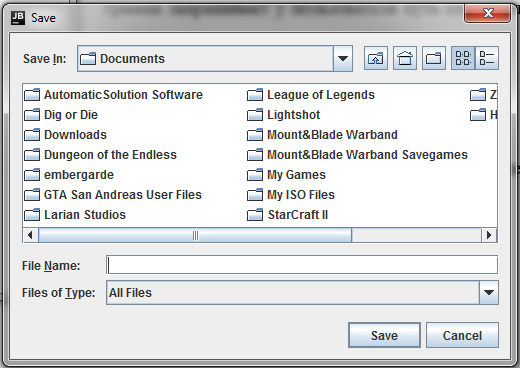


Рисунок 3.2 Сохранение содержимого блокнота в файл.

Также имеется возможность открыть файл с данными. Для этого следует выбрать пункт меню «Файл», затем в выпавшем списке выбрать пункт «Открыть». На экране появится стандартный диалог открытия файла, в котором пользователю будет предложено выбрать расположение открываемого файла. При подтверждении содержимое файла будет прочитано в блокнот (Рисунок 3.3).

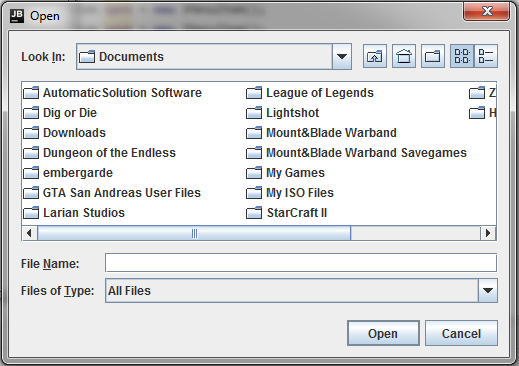


Рисунок 3.3 Открытие файла с данными.

# Заключение

В заключение хочется сказать, что разработка данной программы принесла большое количество опыта в разработке текстовых редакторов.

Достаточно хорошо укрепились навыки работы со средой разработки программного обеспечения «IntelliJ IDEA».

При разработке программы для курсового проекта использовался один из самых популярных языков программирования Java, на котором возможно осуществлять разработку самого разнообразного программного обеспечения: от программ с интерфейсом командной строки до программ с богатым графическим интерфейсом

Были рассмотрены:

* 1. Определение значения слова «Блокнот».
  2. Инструменты, использовавшиеся при разработке проекта.
  3. Написание кода программы.
  4. Описание руководства пользователя.

# 

# Список источников

1. https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.htmlhttps://www.metalcity.ru/info/articles/sklad-kak-element-logisticheskoy-sistemy/istoriya-mirovogo-skladskogo-hozyaistva/sklady-v-rossii/

# Приложение 1

В данном листинге приведён полный весь исходный код программы.

NoteMain.java

**import** java.awt.\*;  
**import** javax.swing.\*;  
**import** java.awt.BorderLayout;  
  
**public class** NoteMain **extends** JFrame { *//Создание главного класса* **public** JEditorPane **textArea** = **new** JEditorPane();*//Добавление компонентов программы* JTextPane **jTextPanel** = **new** JTextPane();

JTextArea **theText** = **new** JTextArea();

JPanel **contentPanel**;

BorderLayout **borderLayout** = **new** BorderLayout();

JMenu **jMenuFile** = **new** JMenu();

JMenuBar **jMenuBar** = **new** JMenuBar();

JLabel **statusBar** = **new** JLabel();

JMenu **File** = **new** JMenu();

JMenuItem **open** = **new** JMenuItem();

JMenuItem **save** = **new** JMenuItem();

JMenuItem **exit** = **new** JMenuItem();

JMenu **View** = **new** JMenu();

JMenuItem **bold** = **new** JMenuItem();

JMenuItem **italic** = **new** JMenuItem();

JMenu **Help** = **new** JMenu();

JMenuItem **popi** = **new** JMenuItem();

**public** NoteMain() **throws** Exception { *//Создание JFrame* **this**.setSize(500, 500);

**this**.setTitle(**"Блокнот"**);

setDefaultCloseOperation(***DISPOSE\_ON\_CLOSE***);

**this**.**textArea**.setFont(**new** Font(**""**, Font.***PLAIN***, 14));

**this**.getContentPane().setLayout(**new** BorderLayout());

**this**.getContentPane().add(**textArea**);

**jMenuBar**.add(**File**);

**jMenuBar**.add(**Help**);

setJMenuBar(**jMenuBar**);  
 }  
 **public static void** main (String args[]) **throws** Exception{  
 Note app = **new** Note();  
 app.setVisible(**true**);  
 }  
  
  
}

Note.java

**import** javax.swing.\*;  
**import** java.awt.event.\*;  
**import** java.io.\*;  
**import** java.util.Scanner;  
**public class** Note **extends** NoteMain {  
 {  
 **File**.setText(**"Фаил"**);

**open**.setText(**"Открыть"**);

**File**.add(**open**);

**save**.setText(**"Сохранить"**);

**File**.add(**save**);

**Help**.setText(**"Помощь"**);

**popi**.setText(**"Подсказка"**);

**Help**.add(**popi**);  
  
  
 **save**.addActionListener(**new** ActionListener() { *//Создание функции save* **public void** actionPerformed (ActionEvent arg0){  
 JFileChooser save = **new** JFileChooser();  
 **int** option = save.showSaveDialog(save);  
 **if** (option == JFileChooser.***APPROVE\_OPTION***) {  
 **try** {  
 BufferedWriter out = **new** BufferedWriter(**new** FileWriter(save.getSelectedFile().getPath()));  
 out.write(**textArea**.getText());  
 out.close();  
 } **catch** (Exception ex) {  
 System.***out***.println(ex.getMessage());  
 }  
 }  
 }  
 });  
  
  
 **open**.addActionListener(**new** ActionListener() { *//Создание функции open* **public void** actionPerformed (ActionEvent arg0){  
 JFileChooser open = **new** JFileChooser();  
 **int** option = open.showOpenDialog(open);  
 **if** (option == JFileChooser.***APPROVE\_OPTION***) {  
 **theText**.setText(**""**);  
 **try** {  
 Scanner scan = **new** Scanner(**new** FileReader(open.getSelectedFile().getPath()));  
 **while** (scan.hasNext())  
 **textArea**.setText(scan.nextLine() + **"\n"**);  
 } **catch** (Exception ex) {  
 System.***out***.println(ex.getMessage());  
 }  
 }  
 }  
 });  
 **popi**.addActionListener(**new** ActionListener() { *//Создание функции popi* **public void** actionPerformed (ActionEvent e){  
 JOptionPane.*showMessageDialog*(**exit**, **"В разработке"**);  
 }  
 });  
  
 }  
 **public** Note() **throws** Exception {  
 }  
  
}