Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

Программное средство редактирования, сравнения текстов и процентной оценки плагиата

БГУИР КР 1-40 01 01 104 ПЗ

Студент: гр. 251001 Василенко И.А.

Руководитель:

асс. Фадеева Е.Е.

Минск 2023

**ВВЕДЕНИЕ**

Редактирование текста — это процесс внесения исправлений, добавлений в текст, изменения его структуры. Цель редактирования — подготовить рукопись по содержанию и форме к изданию. Также целью редактирования текстов является их представление в удобной для читателя форме. В своей работе я попытался создать текстовый редактор именно с этими целями. Текстовые редакторы позволяют просматривать содержимое текстовых файлов и производить над ними различные действия: вставку, удаление и копирование текста, его расположение на странице и т. п.

Проблема плагиата существует уже очень давно. Он получил распространение среди школьников, студентов, аспирантов, докторантов, журналистов с потенциально информационными технологиями. Проблема copy-paste в науке: в среднем 15% рукописей, представляемых в редакции научных журналов, содержат плагиат. В данной работе я попытался создать систему оценки плагиата. Сравнение текстов в данной работе производится с той же целью, что и оценка плагиата.

Данная пояснительная записка содержит следующие основные разделы:

1. Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству;
2. Анализ требований к программному средству и разработка функциональных требований;
3. Проектирование программного средства;
4. Создание (конструирование) программного средства;
5. Тестирование, проверка работоспособности и анализ полученных результатов;
6. Руководство по установке и использованию.
7. **Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству** 
   1. Анализ аналогов к проектируемому программному средству

Существует множество сервисов по оценке плагиата, сравнению и редактированию текстов:

* Text.ru

Text.ru – онлайн-сервис проверки текста на уникальность. Text.ru анализирует материалы объёмом до 15 000 символов. При этом проверка происходит в порядке очереди: почти всегда приходится ждать, пока система обслуживает других пользователей. Систему антиплагиата на Text.ru нельзя обмануть просто изменив порядок предложений, фраз или слов, поменяв падежи, времена или добавив в текст новые слова.

* Microsoft Word

Microsoft Word – редактор текстов. Microsoft Word выполняет все стандартные функции редактирования текстов, такие как создание документа, его сохранение, открытие документа, чтение документа, изменение параметров шрифтов для текста(цвет, размер и др.) , выравнивание текста и т.д. Однако возможности Microsoft Word намного больше. На сегодняшний момент в данной системе присутствуют функции вставки различных формул или нестандартных символов, разметка страницы и размещение текста на ней, добавление ссылок и множество других. Также в Microsoft Word присутствует функция сравнения текстов.

* Advego Plagiatus

Сайт Advego — это целая система сервисов для продажи, заказа и анализа текстов. Инструмент для проверки на уникальность здесь тоже присутствует. На данном сайте текст статьи анализируется сразу по нескольким алгоритмам:

1. Алгоритм шинглов — проверяются точные совпадения фраз, находятся источники копипаста, а также страницы, на которых размещены украденные тексты.
2. Алгоритм лексических совпадений — проверяется схожесть набора лемм, терминов и значимых слов, находятся источники рерайтинга, а также страницы, совпадающие по тематике с проверяемой статьей.
3. Алгоритм псевдоуникализации — проверяется наличие сторонних символов и признаков обработки текста сервисами "повышения" уникальности.

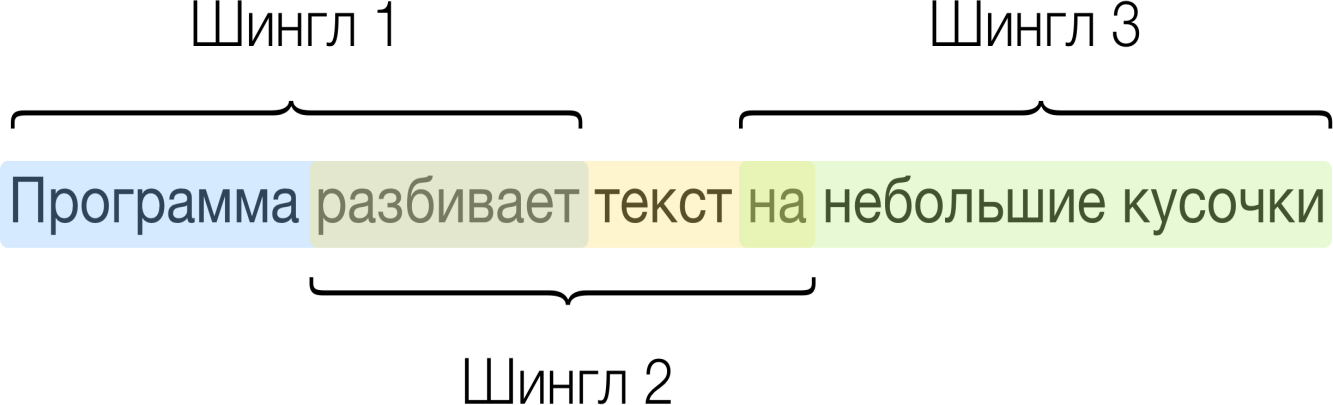
* Google Docs

Google Docs – текстовый редактор. Во многом повторяет функции Microsoft Word.Существуют также такие функции как отправить ссылку на документ письмом прямо из Google Docs, опубликовать документ онлайн или вставить в веб-страницу и т.д.

* 1. Анализ алгоритма оценки плагиата и сравнения текстов

Наиболее известным и часто используемым алгоритмом проверки текста на уникальность является *алгоритм шинглов*.

Шингл — это кусочек текста, размером в несколько слов. Шинглы идут внахлёст друг за другом. По тексту документа можно сформировать множество шинглов, число которых сопоставимо с числом слов в документе.



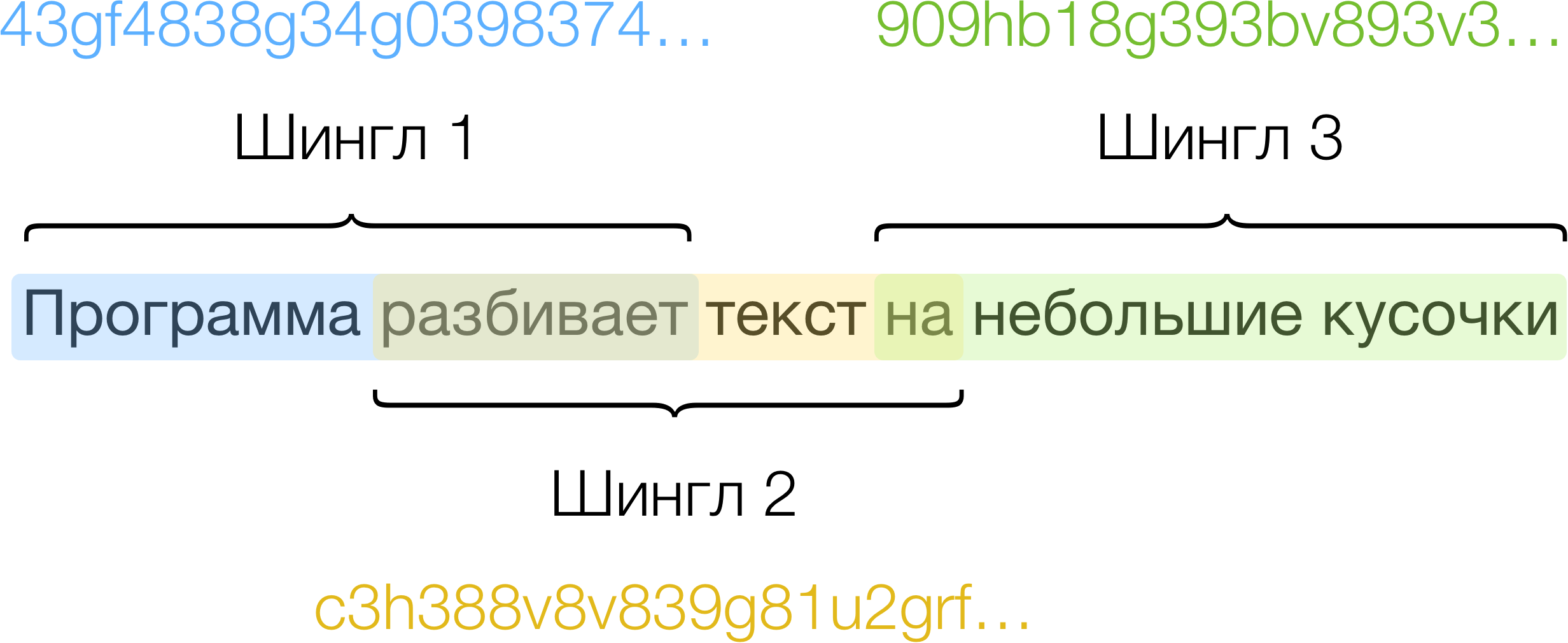
При совпадении нескольких шинглов у двух документов будем считать, что документы пересекаются. Чем больше шинглов совпадает, тем больше одинакового текста в этой паре документов. Индекс занимается поиском документов, обладающих наибольшим количеством пересечений с проверяемым документом.

Индекс шинглов позволяет выполнять две основные операции:

* Индексировать в себе шинглы документов с их идентификаторами:
* Искать в себе и выдавать ранжированный список идентификаторов пересекающихся документов.

Индекс шинглов представляет собой несколько слоёв (массивов) с элементами постоянной длины — от 0 до 128 бит, — которая зависит не только от слоя и не обязательно кратна восьми.

Каждый из слоёв играет определённую роль. Какой-то делает поиск быстрее, какой-то экономит место.



Алгоритм *сравнения текстов*.

Алгоритм выполняет текст будет разбивать тексты на слова. Все буквы, которые имеют одинаковое написание в английском и в русских языках будут приводится к одним буквам( Например a (английское) a (русское) ) Все буквы будут приведены к одному регистру. Будет происходить сравнение слов первого текста со словами второго текста. Если данное слово найдено оно будет выделяться в зависимости от настроек выбранных пользователем. Все неиспользованные слова второго текста будет выделены как “Не найденные”.

* 1. Формирование требований к проектируемому программному средству

Назначение данной разработки – создание текстового редактора, а также средства проверки текста на уникальность.

Входными данными является текст или два текста при сравнении текстов.

В своём проекте, в соответствии с заданием, я буду использовать следующие алгоритмы:

-Упрощенный вариант алгоритма шинглов ( Он будет заключаться в следующем:

1. Текст будет разбиваться на шинглы по три слова
2. Шинглы будут сохраняться в массив
3. Все лишние пробелы будут удаляться и сохраняться в массив будет лишь один пробел между словами
4. Все буквы, которые имеют одинаковое написание в английском и в русских языках будут приводится к одним буквам( Например a (английское) a (русское) )
5. Все буквы будут приведены к одному регистру
6. Сравнение шинглов будет осуществляться соответственно с текстом, хранящимся в отдельном файле

Конечным результатом проверки текста на уникальность будет процентное соотношение количества шинглов совпавших при сравнении к общему количеству шинглов(элементов массива)

- Алгоритм сравнения текстов(пользователь сам выбирает, что он хочет получить в результате. В зависимости от выбора пользователя в результате будут высвечиваться изменения по сравнению со вторым текстом, или будут подсвечиваться символы, которые не изменились по сравнению с первым текстом). Алгоритм заключается в прохождении по символам первого текста и поиском соответствий(или несоответствий) этих символов во втором тексте

Задачей системы является редактирование текста.

Для редактирования текста будут реализованы функции:

* Выравнивание текста по левому, правому краям или по центру
* Изменение параметров шрифта
* Размещение в одной строке не более N символов(N вводится пользователем)

Результатом редактирования текста будет текст.

1. **Анализ требований к ПС и разработка функциональных требований**
   1. Описание функциональных требований к программному средству

Для данного программного средства определены следующие функциональные требования:

* Редактирование текстов
* Возможность изменения шрифта (изменение его стиля, цвета, размера, применение жирного начертания к выделенному тексту, применение курсивного начертания к выделенному тексту, подчеркивание выделенного текста)
* Выравнивание по левому и правому краям, и выравнивание по центру
* Размещение в одной строке не более N символов(N вводится пользователем). Данная функция предполагает следующее:

1. Выбор пользователем количества символов, которое он хочет оставить в строках
2. Выделение текста который он хочет отредактировать
3. Проверка на введенное число(Если N < чем наибольшее слово в выделенном тексте выскакивает соответствующее сообщение)
4. ПС высчитывает количество символов и если N символ остановился на букве, то ПС идет влево по до “не буквы” и переносит символы с номерами большими N на новую строку)

* Сравнение двух текстов предложенных пользователем (пользователь сам выбирает, что он хочет получить в результате. В зависимости от выбора пользователя в результате будут высвечиваться изменения по сравнению со вторым текстом, или будут подсвечиваться символы, которые не изменились по сравнению с первым текстом). Выделение изменений (или совпадений) будет происходить выбранным цветом.
* Функция процентной оценки плагиата(уточнение: данная функция будет работать корректно только с текстом на русском языке(имеется в виду русские слова, символы могут быть записаны латиницей)). Выполнение данной функции описано в п 1.3 (алгоритм шинглов)
* Функция сохранения данного текста в файл(при сравнении текстов сохранится текст первого поля)
* Функция загрузки текста в поле 1
* Функция отмена предыдущего действия
* Поиск подстроки в тексте
  1. Спецификация функциональных требований

1. При запуске ПС появляется поле для текста, над ним появляется панель управления с функциями(основные функции редактирования текста см. п.2.1 редактирование текста, а также кнопки с переходом на функции сравнения текстов и оценки уникальности текста)
2. При нажатии на кнопку сравнения текстов появляется второе поле для второго текста
3. После сравнения текстов чтобы оценить уникальность нужно вернуться на страницу редактирования текста(появится соответствующая кнопка)
4. Для отмены предыдущего действия необходимо перейти по верхней панели. Далее вкладка “Редактирование” и кнопка “Отменить”
5. Для смена цвета текста необходимо: выделить текст на панели управление выбрать цвет и нажать кнопку Change color.
6. Для смены других параметров текста выделяется текст и выбирается соответствующая нужной функции кнопка на панели управления
7. **ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА**

3.1 Разработка хэш-таблицы для эффективного сравнения

Для построения хэш-таблицы используется динамический массив,

каждым элементом которого является однонаправленный список.

При добавлении элемента в хэш-таблицу, высчитывается хэш по хэш- функции. Далее высчитывается его место в таблице, причем :

1. Если на этом месте уже есть элемент, проверяется не совпадают ли наши элементы, если не совпадают переходим к следующему элементу списка, пока не найдем свободное место. Если элемент совпадает с элементом списка просто выходим из функции добавления
2. Если на этом месте нет элемента просто заполняем его нашим элементом

В случае переполнения нашего динамического массива предусмотрена функция Rehash, в соответсвии с которой при заполнении массива на 75% сам массив будет увеличиваться и все элементы перезаписываться в талбицу.

3.2 Разработка алгоритмов сравнения двух текстов

Суть алгоритма заключается в нахождении наибольшей общей последовательности символов(или слов в зависимости от выбора пользователя).

Суть алгоритма приведена ниже

Рассмотрим задачу — для двух данных строк найти такую строку наибольшей длины, которая была бы подпоследовательностью каждой из них. Например, если A=«abcabaac», B=«baccbca» то у строк A и B них есть общая подпоследовательность длины 4, например, «acba» или «acbc».

Рассмотрим последние символы данных строк an-1 и bn-1. Если эти символы совпадают, то они обязательно войдут последними символами и в наибольшую общую подпоследовательность данных строк. Тогда можно свести задачу нахождения наибольшей общей подпоследовательности для строк  A =a0 … an-1  и B =b0 … bm-1  к задаче нахождения наибольшей общей подпоследовательности для строк, полученных отбрасыванием от данных строк последнего символа. Затем к ответу для «укороченных» строк добавим последние (равные) символы исходных строк и получим ответ для исходных строк.

Если же последние символы исходных строк не совпадают, то эти символы не могут одновременно входить в наибольшую общую подпоследовательность, поэтому можно один из них отбросить. Тогда задача сводится к нахождению наибольшей общей подпоследовательности для одного из двух случаев - для строк A =a0 … an-1  и B =b0 … bm-2     или для строк  A =a0 … an-2  и B =b0 … bm-1   .

Мы научились сводить задачу нахождения наибольшей общей подпоследовательности двух строк к меньшей задаче - нахождения наибольшей общей подпоследовательности для строк, полученных отбрасыванием последних символов от исходных строк, то есть для префиксов исходных строк. Для дальнейшего решения задачи будем следовать принципу построения решения при помощи динамического программирования.

Рассмотрим префикс  первой строки из  i символов:  A =a0 … ai-1  и префикс  второй строки из j символов B =b0 … bj-1   .Обозначим через  F(i,j) длину наибольшей общей подпоследовательности.

Теперь выпишем *рекуррентные соотношения*. Они зависят от того, совпадают ли последние символы рассматриваемых строк . Если ai-1 = bj-1  , то тогда F(i,j) = F(i-1,j-1)+1 - нужно решить задачу для строк, полученных отбрасыванием последних символов рассматриваемых строк и добавить 1 символ к ответу. В противном случае нужно рассмотреть два случая: F(i, j-1) и F(i-1, j), которые соответствуют отбрасыванию по одному символу от конца каждой из рассматриваемых строк. В этом случае F(I,j) = max(F(i, j-1) , F(i-1, j)).

Начальные значения функции  задаются просто: если одна из строк - пустая, то общая подпоследовательность также пустая, то есть имеет длину 0 .

Далее необходимо завести двумерный массив размером (n+1) и (m+1) заполнить его значениями по указанным рекуррентным соотношениям. Сначала весь массив заполним нулями (что задаст граничные значения), затем двумя вложенными циклами по  и по  заполним оставшуюся часть массив.

[Источник https://foxford.ru/wiki/informatika/naibolshaya-obschaya-podposledovatelnost]

3.3 Разработка алгоритмов оценки уникальности

Для определения схожести 2-х документов необходимо:

1. канонизация текста;
2. разбиение на шинглы( делим текст по пробелам и создаем массивы.);
3. вычисление хешей шинглов;
4. сравнение, определение результата.

В моем случае канонизация состоит из:

* очистка от лишних пробелов по бокам(trim);
* очистка от таких спец символов ' ” ', ' “ ', ' , ', ' . ' , ' : ', ' $ ', ' # ', ' ( ', ' ) ';
* Все буквы, которые имеют одинаковое написание в английском и в русских языках будут приводится к одним буквам( Например a (английское) a (русское)
* Для надежного опознания слова нам не нужны гласные. Так что следующим этапом сворачивания у нас идёт удаление гласных.

3.4 Алгоритм разбиения на шинглы

Алгоритм заключается в прохождении по строке и выделении слов. После каждого выделенного слова он записывается в массив для шинглов и после этого если массив полон формируется шингл и записывается в хэш-таблицу. После этого все слова в массиве для шинглов сдвигаются вправо.

Схемы основных алгоритмов ПС

Рис.1 Нахождение максимальной длины слова в выделенном тексте

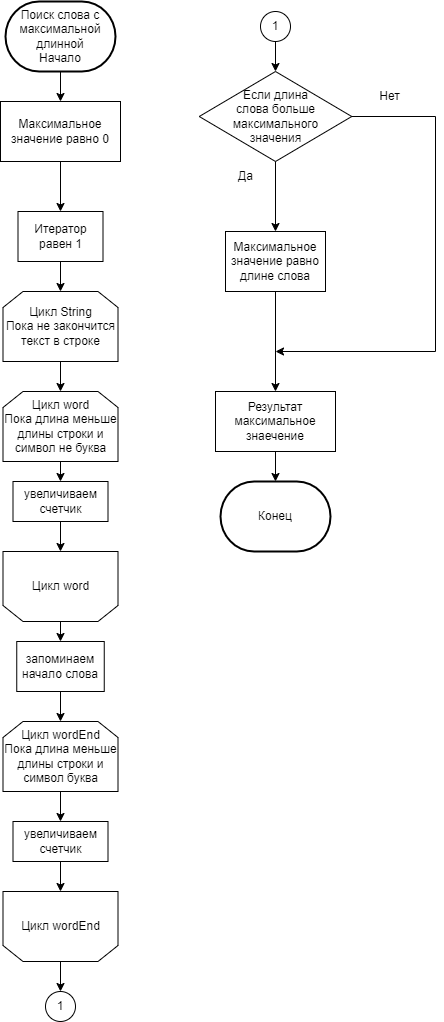


Рис.2 Распределение N количество символов в строке

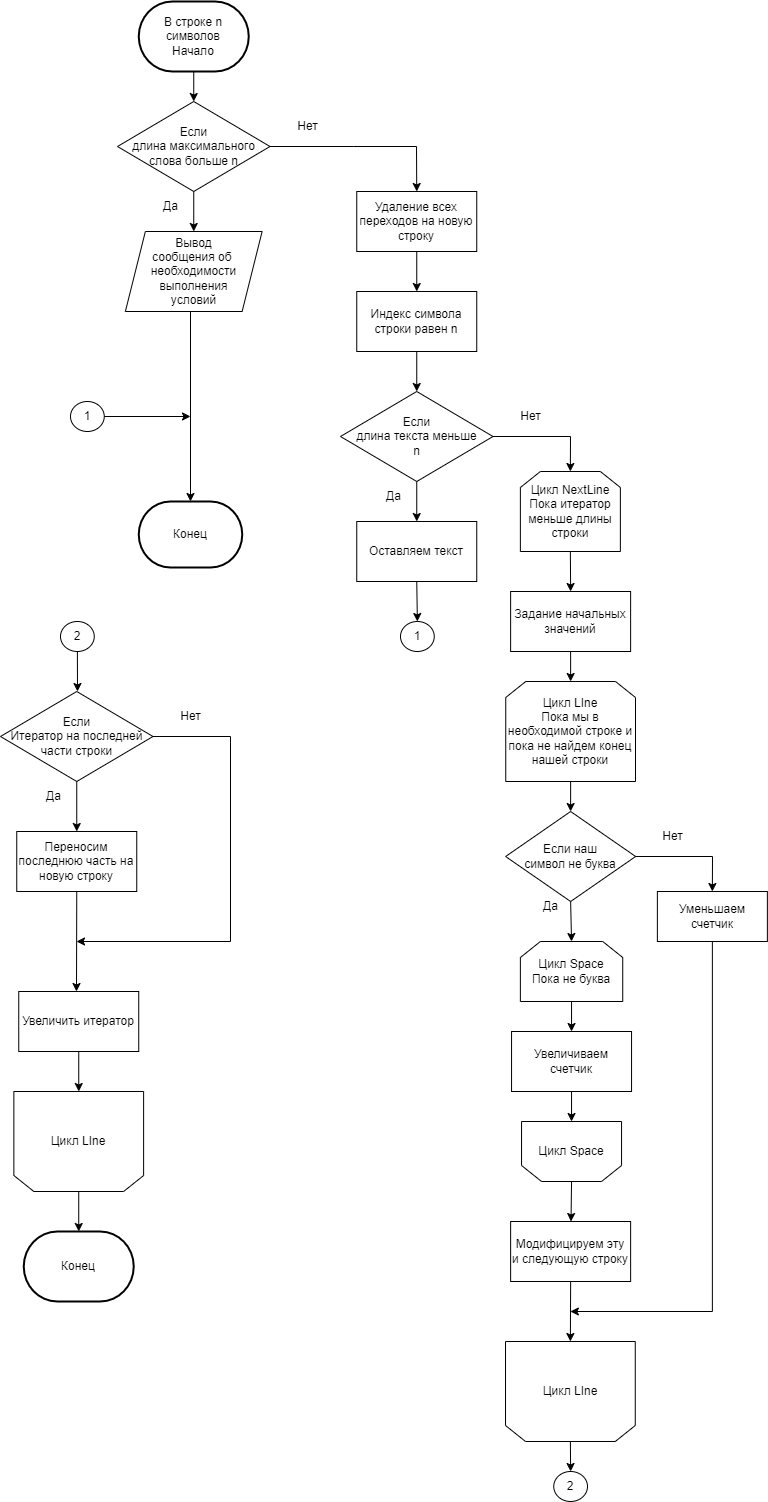


Рис.3 Поиск подстроки

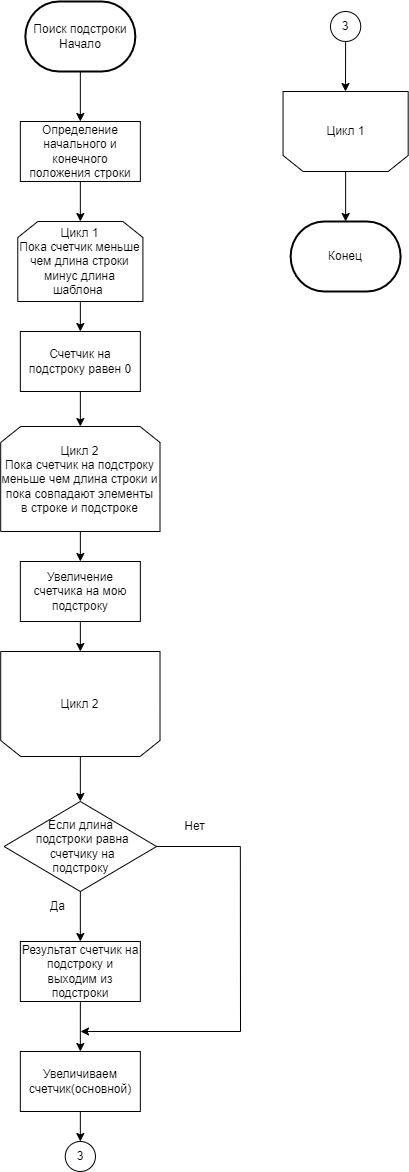


Рис.4 Алгоритм оценки уникальности текста

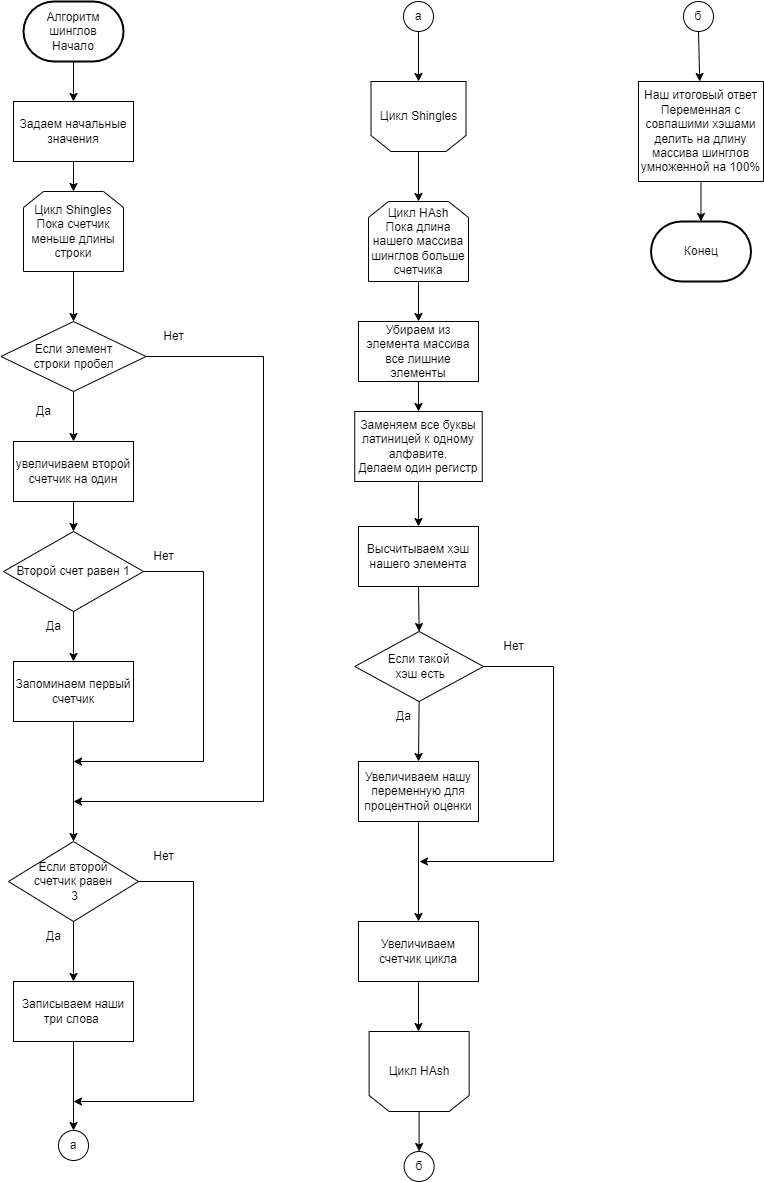
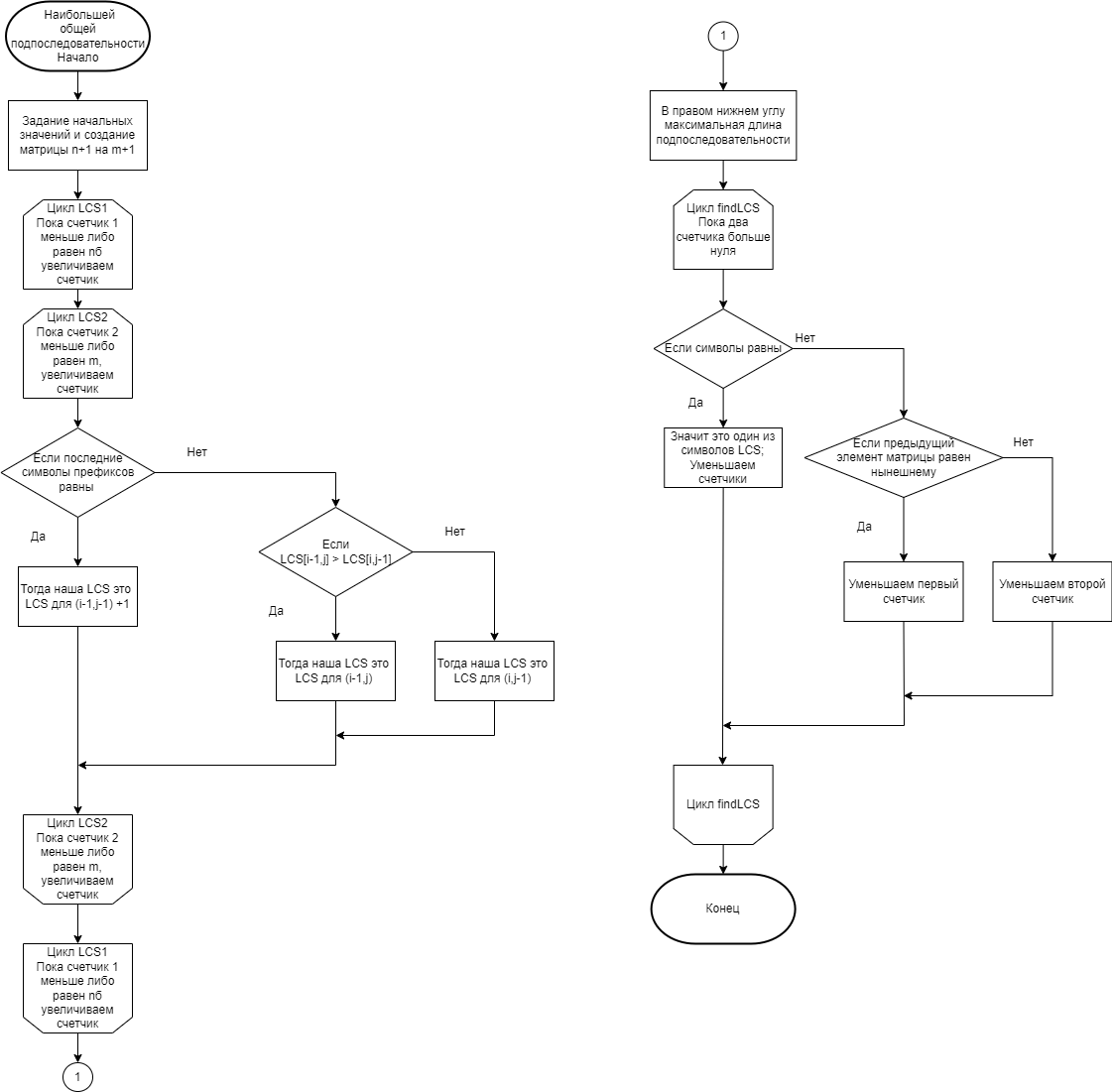


Рис.5 Нахождение наибольшей общей подпоследовательности



**4. СОЗДАНИЕ ПРОГРАММНОГО СРЕДСТВА**

4**.**1 Проектирование модулей программного средства

В ПС используется 4 модуля:

1. Модуль Hash\_Table – модуль с реализацией хэш- таблицы
2. Модуль Redactor – модуль с основным функциям редактирования текста (основной модуль, показывается пользователю по умолчанию)
3. Модуль Compare – здесь находятся все алгоритмы сравнения текстов и все связанные с ними алгоритмы

4**.**2 Разработка модулей программного средства

Таблица 4.2.1 Основные подпрограммы модулей (Hash\_Table)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя подпрог-раммы** | **Описание** | **Заголовок подпрограммы** | **Имя парамет-ра и его тип** | **Назначение параметра** |
| GetHash | Получение хэша строки | function GetHash(const AKey: string): Integer; | A const AKey: string; | строковый ключ |
| CheckMemory | Проверка заполненности хэш-таблицы | function CheckMemory(): boolean; | - | - |
| MemoryAdd | Расширяет нашу таблицу(динамический массив) | procedure MemoryAdd(); | - | - |
| Add | Добавляет элемент в хэш таблицу | procedure Add(const AKey: string; AValue: Integer); | const AKey: string; AValue: Integer | Ключ и его значение |
| Get | Получение номера элемента в массиве по хэшу | function Get(const AKey: string): Integer; | const AKey: string | строковый ключ |
| PutOrUpdate | Добавляет и проверяет нужно ли расширять таблицу | procedure PutOrUpdate(const AKey: string; AValue: Integer); | const AKey: string; AValue: Integer | Ключ и его значение |
| ContainsKEy | Проверяет содержит ли данный ключ хэш таюлица | function ContainsKEy(const AKey: string): boolean; | const AKey: string | строковый ключ |

Таблица 4.2.2 Основные подпрограммы модулей (Redactor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя подпрог-раммы** | **Описание** | **Заголовок подпрограммы** | **Имя парамет-ра и его тип** | **Назначение параметра** |
| OpenBtnClick | Обработчик нажатия кнопки открыть файл | procedure OpenBtnClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| SaveASClick | Обработчик нажатия кнопки Сохранить как файл | procedure SaveASClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| BoldButton1Click | Обработчик нажатия кнопки сделать жирным шрифты | procedure BoldButton1Click(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| ItalicButton1Click | Обработчик нажатия кнопки сделать курсивом шрифты | procedure ItalicButton1Click(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| UnderlineButton1Click | Обработчик нажатия кнопки сделать подчеркнутым шрифты | procedure UnderlineButton1Click(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| LeftJustifyButton1Click | Обработчик нажатия кнопки выровнять по левому краю | procedure LeftJustifyButton1Click(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| CenterJustifyButton1Click | Обработчик нажатия кнопки выровнять по центру | procedure CenterJustifyButton1Click(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| RightJustifyButton1Click | Обработчик нажатия кнопки выровнять по правому краю | procedure RightJustifyButton1Click(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| FontButton1Click | Обработчик нажатия кнопки открыть диалоговое окно со шрифтами | procedure FontButton1Click(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| UndoClick | Обработчик нажатия кнопки отменить | procedure UndoClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |

Продолжение таблицы 4.2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SignsNumberClick | Обработчик нажатия на кнопку для формирования строк по заданному количеству символов | procedure SignsNumberClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| FormCreate | Обработчик создания формы | procedure FormCreate(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| CompareUnitButtonClick | Обработчик нажатия на кнопку для перехода к форме сравнения | procedure CompareUnitButtonClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| ColorBox1Select | Обработчик нажатия на кнопку для выбора цвета шрифта | procedure ColorBox1Select(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| CreateNewClick | Обработчик нажатия кнопки создать новый файл | procedure CreateNewClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| ExitButtonClick | Обработчик нажатия кнопки выйти из приложения | procedure ExitButtonClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| SaveClick | Обработчик нажатия кнопки Сохранить файл | procedure SaveClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| CopyBtnClick | Обработчик нажатия кнопки копировать выделенное | procedure CopyBtnClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| InsertBtnClick | Обработчик нажатия кнопки вставить текст из буфера | procedure InsertBtnClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| SelectAllBtnClick | Обработчик нажатия кнопки выделить весь текст | procedure SelectAllBtnClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| procedure CutBtnClick(Sender: TObject); | Обработчик нажатия кнопки вырезать текст | procedure CutBtnClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| procedure SAveFile; | Сохраняет файл | procedure SAveFile; |  |  |

Продолжение таблицы 4.2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FontSizeEdit1KeyPress | Обработчик нажатия клавиши увеличить шрифт | procedure FontSizeEdit1KeyPress(Sender: TObject; var Key: Char); | Sender: TObject | Объект, который сгенерировал событие |
| UpDown1Changing | Обработчик нажатия кнопки изменения размера шрифта | procedure UpDown1Changing(Sender: TObject; var AllowChange: Boolean); | Sender: TObject | Объект, который сгенерировал событие |
| UpDown2Changing | Обработчик нажатия кнопки изменения количества символов строке | procedure UpDown2Changing(Sender: TObject; var AllowChange: Boolean); | Sender: TObject; var AllowChange: Boolean); | Объект, который сгенерировал событие |
| NumberSignsEdit1KeyPress | Обработчик нажатия кнопки формирования строк заданной длины | procedure NumberSignsEdit1KeyPress(Sender: TObject; var Key: Char); | Sender: TObject; var Key: Char | Объект, который сгенерировал событие |
| FindButtonClick | Обработчик нажатия кнопки поиска символов | procedure FindButtonClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| procedure FindEditKeyPress | Обработчик нажатия клавиши поиска символов | procedure FindEditKeyPress(Sender: TObject; var Key: Char); | Sender: TObject; var Key: Char | Объект, который сгенерировал событие |
| Splitter1Moved | Обработчик движения сплиттера для регулировки отступа | procedure Splitter1Moved(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| FindMax; | Поиск длины самого длинного слова | function FindMax(str: string): integer; | str: string | Строка для поиска |

Таблица 4.2.3 Основные подпрограммы модулей (Compare)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| procedure CompareText | Процедура сравнения текстов | procedure CompareText(mode: integer); | mode: integer | Режим вывода |
| CompareTextsClick | Обработчик нажатия кнопки сравнения текстов | procedure CompareTextsClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| ShinglesComboSelect | Обработчик нажатия combobox для определения шинглов | procedure ShinglesComboSelect(Sender: TObject); | (Sender: TObject); | Объект, который сгенерировал событие |
| GoBAckbuttonClick | Обработчик нажатия кнопки вернуться в текстовый редактор | procedure GoBAckbuttonClick(Sender: TObject); | Sender: TObject; | Объект, который сгенерировал событие |
| ReplaceBtnClick | Обработчик нажатия кнопки поменять тексты местами | procedure ReplaceBtnClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| PanelU3ButtonsResize | Обработчик изменения панели | PanelU3ButtonsResize(Sender: TObject); | Sender: TObject; | Объект, который сгенерировал событие |
| signsRadioButtonClick | Обработчик нажатия кнопки выбора режима вывода результатов | procedure signsRadioButtonClick(Sender: TObject); | Sender: TObject | Объект, который сгенерировал событие |
| WordsRadioButtonClick | Обработчик нажатия кнопки выбора режима вывода результатов | procedure WordsRadioButtonClick(Sender: TObject); | Sender: TObject | Строка для поиска |
| OneWord | Разделяет строку на слова или символы в зависимости от режима | function OneWord(var str1: string; mode: integer): ArrOfStr; | var str1: string; mode: integer | Строка и режим вывода |
| MarkDiffer | Разметка удаляем(добавляемых) слов (строк) | function MarkDiffer(strNUms1: ArrOfStr; LCS: ArrOfStr): ArrOfStr; | strNUms1: ArrOfStr; LCS: ArrOfStr | Массив слов и наибольшая общая подпоследовательность |
| MaxSubLine | Наибольшая общая подпоследовательность двух массивов | function MaxSubLine(strNUms1: ArrOfStr; strNUms2: ArrOfStr): ArrOfStr; | strNUms1: ArrOfStr; strNUms2: ArrOfStr | Массив слов первого текста и второго |

Продолжение таблицы 4.2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| lowCase | Приведение к нижнему регистру | function lowCase(var str1: string): string; | var str1: string | Строка |
| Shingles | Разбиение текста на шинглы | function Shingles(str1: string; n: integer): THAshTAble; | str1: string; n: integer | Сторка и количество слов в шингле |
| Count | Подсчет процента схожести текстов | function Count(HashTable: THAshTAble; str1: string; n: integer): integer; | HashTable: THAshTAble; str1: string; n: integer | Таблица слов текста сравнения и количество слов в шингле |

**5. ТЕСТИРОВАНИЕ, ПРОВЕРКА РАБОТОСПОСОБНОСТИ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ**

Таблица 5.1 Тестирование работоспособности ПС

|  |  |  |  |
| --- | --- | --- | --- |
| Номер  теста | Тестируемая функциональность | Ожидаемый результат | Полученный результат |
| 1 | Сохранение текста редактирования | Текст успешно сохранен | Тест пройден |
| 2 | Открытие файла для редактирования | Файл успешно открывается |  |
| 3 | Сохранить текст в файл из которого был открыт текст(перезапись файла) | Файл успешно перезаписан | Тест пройден |
| 4 | Создание нового текста | Очищение панели редактирования | Тест пройден |
| 5 | При создании нового текста появление окна, если файл не был сохранен | Появление окна сохранения  С выбором сохранять изменения в файл или нет или отменить действие | Тест пройден |
| 6 | Копирование и вырезка текста на горячие клавиши, на кнопки | Копирование текста в буфер | Тест пройден |
| 7 | Вставка текста из буфера на горячие клавиши, на кнопки | Вставка текста из буфера | Тест пройден |
| 8 | Выделение всего текста и отмена действий | Выделение текст и отмена |  |
| 9 | Редактирование отступа в тексте при помощи ползунка | Успешная регулировка отступов | Тест пройден |
| 10 | Формирование строк с заданным количеством символов | Отредактированный в соответствии с функцией текст |  |
| 11 | Изменение параметров шрифта | Изменение выделенного текста | Тест пройден |
| 12 | Поиск и выделение слова в тексте | Выделение найденного текста  Или появление сообщения о не найденном тексте | Тест пройден |
| 13 | Выход из программы | Закрытие окон | //Тест пройден |
| 14 | Переход и возвращение на окна редактирования и сравнения | Переход между формами | Тест пройден |
| 15 | Смена текста местами (в модуле сравнения) | Текст поменялся местами | Тест пройден |
| 16 | Сравнение двух текстов при режиме ‘Символы’ | В результате текст с подсвеченными результатами текста | Тест пройден |
| 17 | Сохранение результатов сравнения | Файл с сохраненными данными |  |
| 18 | Оценка схожести текстов | Процент схожести текстов | Тест пройден |
| 19 | Сравнение двух текстов при режиме ‘Слово’ | В результате текст с подсвеченными результатами текста | Тест пройден |

5.2 Анализ полученных результатов

Тест

|  |  |
| --- | --- |
| Текст 1 | Текст 2 |
| Главный герой — 27-летний Илья Горюнов, семь лет отсидевший в тюрьме по ложному обвинению в распространении наркотиков. Когда Илья выходит на свободу, он понимает, что прежняя жизнь, по которой он тосковал, разрушена, и вернуться к ней он больше не сможет. Хотя он не собирался мстить человеку, который отправил его в тюрьму, другого выхода теперь нет. Встретившись лицом к лицу со своим обидчиком, Илья совершает необдуманный поступок, после которого главный герой получает доступ к смартфону Петра и Ивана, а с ним и к жизни молодого человека — его фотографиям и видео, перепискам с родителями и девушкой Ниной, к странным | Главный герой — 7-летний Илья Горюнов, семь лет отсидевший в тюрьме по ложному обвинению в распространении наркотиков. Когда Илья выходит на свободу, он понимает, что прежняя жизнь, по которой он тосковал, разрушена, и вернуться к ней он больше не сможет. Хотя он не собирался мстить человеку, который отправил его в тюрьмы, другого выхода теперь нет. Встретившись лицом к лицу со своими обидчиками, Илья совершает необдуманный поступок, после которого главный герой получает доступ к смартфону Петра, а с ним и к жизни молодого человека — его фотографиям и видео, перепискам с родителями и девушкой Ниной, к странным |

|  |
| --- |
| *Результат*  Главный герой — 27-летний Илья Горюнов, семь лет отсидевший в тюрьме по ложному обвинению в распространении наркотиков. Когда Илья выходит на свободу, он понимает, что прежняя жизнь, по которой он тосковал, разрушена, и вернуться к ней он больше не сможет. Хотя он не собирался мстить человеку, который отправил его в тюрьмуы, другого выхода теперь нет. Встретившись лицом к лицу со своими обидчикоами, Илья совершает необдуманный поступок, после которого главный герой получает доступ к смартфону Петра и Ивана, а с ним и к жизни молодого человека — его фотографиям и видео, перепискам с родителями и девушкой Ниной, к странным |

Результат схожести двух текстов

