

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина: Программирование

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

Текстовой редактор

Выполнил: студент: гр. 653501

Юшкевич А.Ю.

Руководитель: ассистент кафедры
информатики Козуб В.Н.

Минск 2017

Оглавление

Постановка задачи	3
Разработка программы. Работа с файлами	4
Разработка программы. Работа с текстом	6
Заключение	8
Список использованной литературы	9

Постановка задачи

Целью курсовой работы стало создание текстового редактора, позволяющего легко работать с большинством файлов различных форматов, но сохраненных в стандартных кодировках.

Редактор должен выполнять следующие функции:

1. Работа с файлами
 - Открытие файла и считывание данных из него
 - Сохранение данных в файл
 - Создание нового файла и работа с ним
2. Работа с текстом
 - Поиск ключевого слова в тексте
 - Изменение размера шрифта
 - Изменение цвета текста
3. Вывод информации о программе
4. Возможность диалога с пользователем на разных языках

Разработка программы

Работа с файлами

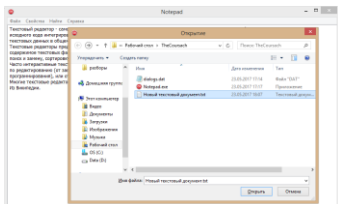
При запуске программы открывается окно, в котором сразу можно начинать работать.



Пример интерфейса

Помимо основной рабочей области присутствуют элементы меню.

1. Компонент Файл позволяет взаимодействовать с файлами



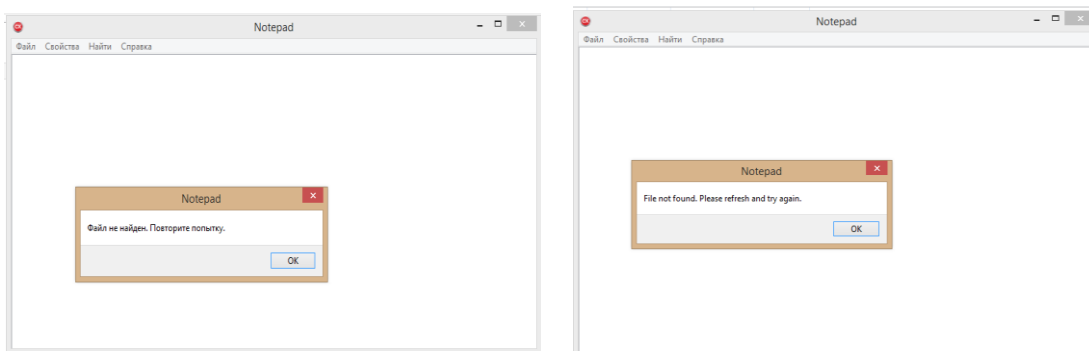
При этом есть возможность открытия файлов не только формата .txt, но и других файлов. Например, .cpp, .h, .pas, .ini, .inf, .log и т.д. В этом случае программа может выступать в качестве редактора кода. Аналогично присутствует возможность сохранения файла в любом формате, включая пользовательские, например, .abc, .qwerty, с последующим их открытием.

Существует возможность сохранять один файл под несколькими именами и в разных форматах. Полностью поддерживается совместимость с такими популярными редакторами как notepad++ и блокнот Windows.

Работа с файлами реализована с помощью набора методов и классов fstream. Далее код загрузки файла.

```
bool Load(char *fileName)    {  
    ifstream fin("fileName");  
    if (!fin.is_open())  
    {  
        ShowMessage( dialogs[ errorMessage ] );  
        return false;  
    }  
    //...  
    fin.close();  
}
```

В случае попытки открытия несуществующего файла будет выведено сообщение об ошибке на русском языке, но существует возможность изменения языка диалогов на любой другой, это реализовано с помощью выноса диалоговых сообщений в отдельный файл(dialogs.dat).

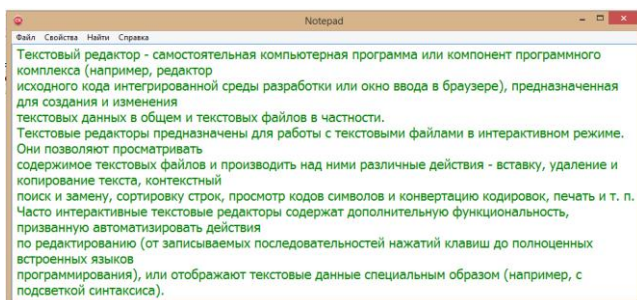


Также возможна реализация полной локализации приложения и изменение языка в процессе работы.

Работа с текстом

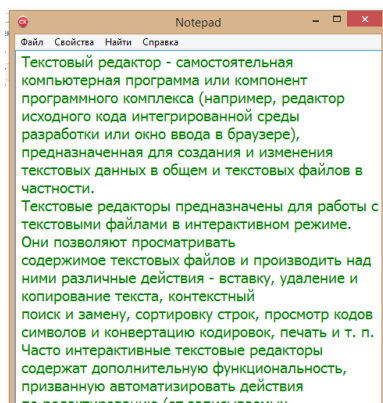
Базовые функции взаимодействия с редактором(ввод символов с клавиатуры, удаление символов, копирование вставка) реализованы с помощью класса TMemo VCL(*Visual Component Library*).

Пользователь может самостоятельно задавать свойства такие свойства текста, как размер шрифта, его цвет.



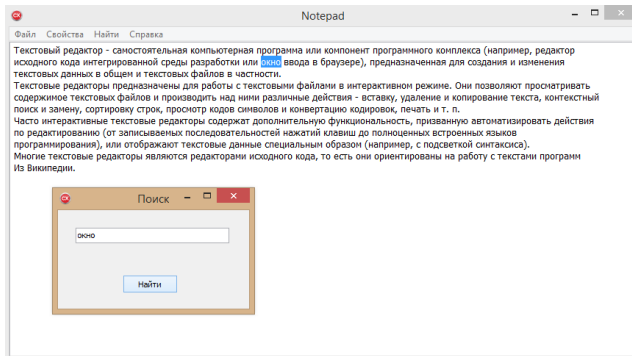
Пример

Присутствует возможность автоматического изменения строения текста в случае изменения размеров окна. Данная функция является преимуществом редактора N.O.T.E.P.A.D. перед другим популярным редактором.



Тот же текст, но с измененными размерами

Пользователь имеет возможность искать ключевые слова или словосочетания в тексте, при это будет найдено количество совпадений и выделено первое такое совпадение.



Поиск осуществляется с помощью алгоритма Кнута-Морриса-Пратта(КМП), что позволяет делать это эффективно даже для достаточно больших файлов. Алгоритм основан на префикс-функции, подробнее об этом можно прочитать в книге "Алгоритмы. Построение и анализ"(источник 2). Далее приводится реализация КМП.

```
void KMP(AnsiString text, int subLength, int &number)
{
    std::vector<int> pF(text.Length(), 0);
    for (int i = 1; i < pF.size(); ++i)
    {
        int j = pF[i - 1];
        while (i != j && text[i] != text[j])
            j = text[j - 1];

        pF[i] = j + (text[i] == text[j] ? 1 : 0);
        if (pF[i] == subLength)        ++number;
    }
}
```

Заключение

При выполнении курсовой работы был реализован текстовый редактор, дающий пользователю набор инструментов для просмотра и редактирования файлов.

Были реализованы алгоритмы работа с текстовыми данными, использованы класс `fstream` и `TMemo`.

Исходный код программы предусматривает возможности улучшения и расширения функционала текстового редактора, например, возможно добавление подсветки синтаксиса для языков программирования, открытия файлов специфических форматов(.docx, .zip, .hex).

Полный исходный код доступен на

github.com/programming-653501/Yushkevich-A/tree/master/CourseWork/Source

Список литературы:

1. Тьюториал по работе с C++ builder [Электронный ресурс] - Режим доступа: <http://docwiki.embarcadero.com/RADStudio/XE7/en/Tutorials>
Дата доступа: 23.05.2017
2. Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И. В. Красикова. — 2-е изд. — М.: Вильямс, 2005. — 1296 с. — [ISBN 5-8459-0857-4](#).
3. Тьюториал по работе с fstream [Электронный ресурс] - Режим доступа: <msdn.microsoft.com/en-us/library/k8w40w6t.aspx>
Дата доступа: 23.05.2017