Частное учреждение образования

«Колледж бизнеса и права»

ПРОГРАММА ДЛЯ ОБРАБОТКИ ТЕКСТОВЫХ ФАЙЛОВ С ФУНКЦИЕЙ АНТИПЛАГИАТА

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

«Основы алгоритмизации и программирование»

КП Т.0910015.401

Руководитель проекта (Ю.В. Шаляпин)

Учащийся (А.С. Самойлов)

2022

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Датаа*

*Лист*

*3*

*КП Т.0910015.401*

*Разраб.*

*Самойлов А.С.*

*Провер.*

*Шаляпин Ю.В.*

*Реценз.*

*Н. Контр.*

*Утверд.*

*Программа для обработки текстовых файлов с функцией антиплагиата*

*Лит.*

*Листов*

*КБиП*

55

*у*

Содержание

[Введение 4](#_Toc107267525)

[1 Объектно-ориентированный анализ и проектирование системы 5](#_Toc107267526)

[1.1 Сущность задачи 5](#_Toc107267527)

[2 Вычислительная система 6](#_Toc107267528)

[2.1 Требования к аппаратным и операционным ресурсам 6](#_Toc107267529)

[2.2 Инструменты разработки 6](#_Toc107267530)

[3 Проектирование задачи 7](#_Toc107267531)

[3.1 Требования к приложению 7](#_Toc107267532)

[3.2 Концептуальный прототип 7](#_Toc107267533)

[3.3 Организация данных 9](#_Toc107267534)

[3.4 Функции и элементы управления 11](#_Toc107267535)

[3.5 Проектирование справочной системы 11](#_Toc107267536)

[4 Описание программного средства 13](#_Toc107267537)

[4.1 Общие сведения 13](#_Toc107267538)

[5 Методика испытаний 14](#_Toc107267539)

[5.1 Технические требования 14](#_Toc107267540)

[5.2 Функциональное тестирование 14](#_Toc107267541)

[6 Применение 16](#_Toc107267542)

[6.1 Назначение программы 16](#_Toc107267543)

[6.2 Условия применения 16](#_Toc107267544)

[6.3 Справочная система 16](#_Toc107267545)

[Заключение 17](#_Toc107267546)

[Список информационных источников 18](#_Toc107267547)

[Приложение А. Текст программных модулей 18](#_Toc107267548)

[Приложение Б. Результат работы программы 55](#_Toc107267549)

Приложение В. Блок схемы работы алгоритма антиплагиата 56

Введение

Целью курсового проекта является создание программы с реализацией обработки тестовых файлов с функцией антиплагиата.

Для выполнения поставленной цели необходимо решить следующие задачи: проанализировать программы аналоги, выявить у них плюсы и минусы и учесть это в своем программном средстве, разработать код на С++, а также протестировать основной функционал программного средства.

В условиях информационной вседоступности стало сложно контролировать количество авторского творчества. В данное время антиплагиат является очень важным средством для проверки авторского текста, который может использовать для проверки написанных докладов, курсовых и т. д. Данное программное средство может использоваться в разных сферах труда. Целью данной курсовой работы является создание программного средства для редактирования текста, нахождения процента сходства между двумя текста.

Актуальность проведения работы является большое количество текстов, которые не являются авторскими, однако их выдают за авторские.

При анализе программ аналогов таких как: Extx, Advego Plagiatus, Antiplagarism Net 4.1. Данные программы имею такие плюсы как:

* наличие баз данные с текстами для сравнения;
* кроссплатформенность;
* наличие разных типов файлов для сравнения;
* обнаружение замены синонимами.

Так же при использовании программ были обнаружены таким минусы как:

* платная проверка текста;
* отсутствие оффлайн-версии;
* сложность в освоении программы.

В первом разделе «Объектно-ориентированный анализ и проектирование системы» описывается сущность поставленной задачи, исследование предметной области. Так же описываются основные функции программы.

Во втором разделе «Вычислительная система» описывается аппаратные и операционные характеристики компьютера, программное обеспечение, язык программирования на котором разрабатывалось программное средство.

В третьем разделе «Проектирование задачи» описываются основные требования к программному средству, требования к графическому интерфейсу. Так же в данном разделе будет описан каждый элемент интерфейса.

В четвертом разделе «Описание программного средства» будут перечислены подключенные модули приложения, программные средства необходимые для работы программы, описание инсталляции программного средства.

В пятом разделе «Методика испытаний» описывается перечень требований к техническим средствам для проведения тестирования.

В шестом разделе «Применение» происходит описание сведений о назначении программного средства и области его применения, ограничениях. Так же описывается справочная система, средства разработки и ее структура.

В заключении будут представлены выводы о созданном программном приложении.

1. Объектно-ориентированный анализ и проектирование системы
   1. Сущность задачи

Плагиат – умышленно совершаемое физическим лицом незаконное использование или распоряжение охраняемыми результатами чужого творческого труда, которое сопровождается доведением до других лиц ложных сведений о себе как о действительном авторе. Плагиат может быть нарушением авторско-правового законодательства и патентного законодательства и в качестве таковых может повлечь за собой юридическую ответственность, как гражданскую, так и в отдельных случаях даже уголовную.

Данное программное средство ускоряет процесс проверки текстов между собой на процент сходства, так как программа анализирует текст быстрее и надежнее чем человек.

Данное программное средство может быть полезно в любой отрасли труда. Так как плагиат может встречаться везде.

В ходе выполнения курсового проекта необходимо создать программу, в которой пользователь сможет редактировать текст, а также алгоритм, который будет проверять текст наличие антиплагиата. В программе должны быть реализованы следующие функции:

* редактирование текста;
* форматирование текста;
* создание и сохранение текстовых файлов;
* открытие и чтение текстовых файлов.

В данное время существует множество аналогов программного средства, однако в некоторых программах обнаружены некоторые проблемы.

1. Вычислительная система
   1. Требования к аппаратным и операционным ресурсам

Во время разработки программного средства был использован персональный компьютер со следующими характеристики:

* процессор Intel Pentium B950, 2011 года выпуска, техпроцесс 32 нанометра, частота 2,10 ГГц, 2 ядра, 2 потока;
* оперативная память 6 ГБ, DDR3, 1333 Гц;
* операционная система MS Windows, 64 разрядная;
* видеокарта NVIDIA GeForce 315M 512 МБ;
* мышь SMARTBUY SBM-340AG-K.
  1. Инструменты разработки

В ходе разработки программного средства были использованные следующие инструменты:

* операционная система Windows 10;
* среда программирования MS Visaul Studio 2019 Community;
* язык программирования C++.

Windows – группа семейств коммерческих операционных систем корпорации Microsoft. Данная операционная система является универсальной и может использовать для разных целей. Данная операционная система позволяет пользователю компьютера работать с файлами, пользоваться Интернетом, просматривать видеозаписи, прослушивать музыку и многое другое. В данное время операционные системы Windows являются самыми распространенными операционными системами в мире.

Программное средство было разработана на языки C++ с использование библиотек WindowsForms для создания окон программы. С++ - является компилируемый, статически типизированный язык программирования общего назначения.

Причинами выбора языка C++ стали его быстродействие в выполнении поставленных задач и желание улучшить навык владения данным языком программирования.

Выбором библиотек WindowsForms послужила ее простота в освоении, понятный к восприятию интерфейс.

Среда программирования MS Visual Studio 2019 Community также является продуктом компании Microsoft. Она обеспечивает комфортную разработку, тестирование и компиляцию прикладных программ.

Возможность сетевой поддержки не предусматривается.

1. Проектирование задачи
   1. Требования к приложению

Данное программное средство должно удовлетворять следующему комплексу требований:

* лаконичный и понятный пользовательский интерфейс;
* локализация на 3 языках;
* быстрота действия;
* многократное использование данных;
* защита данных (авторизация).
  1. Концептуальный прототип

Разработанное программное средство представляет собой файл «AntiplagiarismFix.exe».

При двойном нажатии правой кнопкой мыши на файл, запускается окно аутентификации, состоящее из полей для ввода данных, вкладка «Настройки» и 2 кнопки управления, изображенных на рисунке 1.

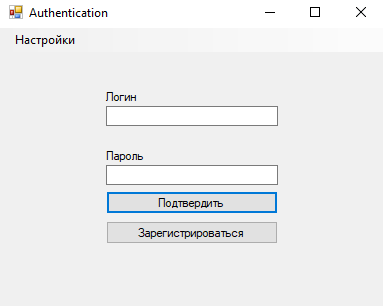


Рисунок 1 – Окно аутентификации

В данном окне имеются 2 строки для ввода данных, 2 кнопки и вкладка «Настройки». В строки вводятся логин и пароль пользователя, который пользуется программой. В вкладке «Настройки» пользователь может изменять язык программы или изменить цвет заднего фона. При нажатии на кнопку «Зарегистрироваться» откроется окно регистрации, изображенном на рисунке 2.

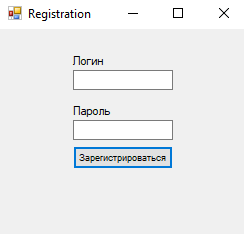


Рисунок 2 – Окно регистрации

Окно регистрации имеет 2 строки для ввода данных и кнопка. При нажатии на кнопку «Зарегистрироваться» происходит создание нового пользователя и запись его данных в базу данных для последующего входа в аккаунт.

Вернемся к окну аутентификации и рассмотрим кнопку «Подтворить», при нажатии на которую происходит скрытие формы аутентификации и открытие главной формы, изображенной на рисунке 3.

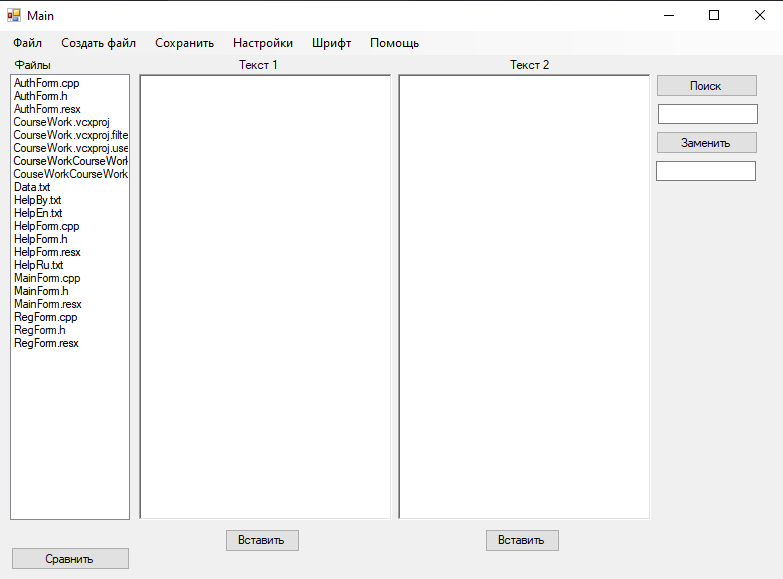


Рисунок 3 – Главное окно

В данном окне пользователь встречает 4 строки для ввода данных ,5 кнопок, а также 6 вкладок. При нажатии на кнопку «Сравнить» происходит сравнение двух текстов, которые расположены в поле «Текст 1» и «Текст 2» после расчета программы выведет результат около кнопки «Сравнить».

При заполнении поля под кнопкой «Поиск» и нажатии кнопку «Поиск» будет произведет поиск введённого текста в поле «Текст 1» и «Текст 2».

При заполнении поля под кнопкой «Поиск» и поля под кнопкой «Заменить» и нажатии кнопку «Заменить» будет произведет поиск введённого текста в поле «Текст 1» и «Текст 2» и его замена на новый текст вместо выбранного.

При нажатии на вкладку «Помощь» откроется окно справки, в котором находится краткая информация о программе, справка изображена на рисунке 4.

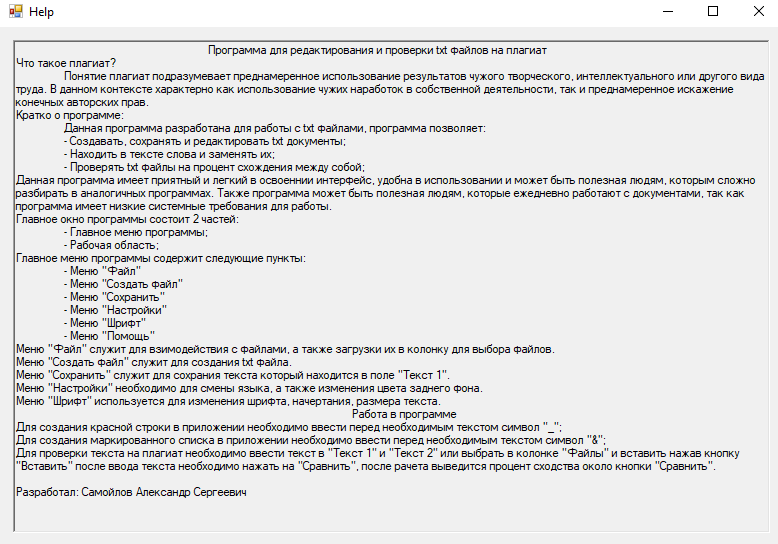


Рисунок 4 – Окно справки

* 1. Организация данных

Данные, логин и пароль пользователей располагаются в файле «Data.txt». Остальные данные инициализируются во время запуска программы и располагаются в выбранном пользователем месте.

В этом подразделе главная задача состоит в том, чтобы рассмотреть основные функции программы с описанием назначения каждой кнопки. Элементы управления и визуализации расположены в таблице 1.

Таблица 1 – Элементы управления и визуализации

| Размещение | Элемент экрана | Класс | Действие |
| --- | --- | --- | --- |
| Главная форма | Вставить | Button | Вывод текста из выбранного файла в текстовое поле |
| Найти | Button | Поиск слова в тексте 1 |
| Заменить | Button | Заменяет слово в тексте, на которое ввел пользователь |
| Сравнить | Button | Нахождение процента сравнения двух текстов |
| Файл | ToolStripMenuItem | Открытие файла и запись данных из него в память |
| Файл – Открыть | ToolStripMenuItem | Открытие файла и запись данных из него в память |
| Создать файл | ToolStripMenuItem | Создание файла |
| Сохранить | ToolStripMenuItem | Открытие окна для сохранения файла |
| Настойки | ToolStripMenuItem | Открытие настроек для изменения цвета и языка |
| Настойки – Задний фон | ToolStripMenuItem | Выбор цвета заднего фона |
| Настойки – Язык | ToolStripMenuItem | Выбор языка программы |
| Шрифт | ToolStripMenuItem | Открытие настроек для изменения шрифта |
| Сохранить | ToolStripMenuItem | Сохранение файла в файл |
| Файл – Сохранить как… | ToolStripMenuItem | Сохранение графа в новый файл |
| Помощь | ToolStripMenuItem | Открытие краткой информации об авторе и программе |
| Файлы | fileListBox | Выбор файла для открытия |
| Аутентификация | Подтвердить | Button | Переход на форму «Главная» |
| Зарегистрироваться | Button | Переход на форму «Регистрация» |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | Настойки | ToolStripMenuItem | Открытие настроек для изменения цвета и языка |
| Настойки – Задний фон | ToolStripMenuItem | Выбор цвета заднего фона |
| Настойки – Язык | ToolStripMenuItem | Выбор языка программы |
| Зарегистрироваться | Button | Регистрация пользователя |
| Настойки | ToolStripMenuItem | Открытие настроек для изменения цвета и языка |
| Настойки – Задний фон | ToolStripMenuItem | Выбор цвета заднего фона |
| Настойки – Язык | ToolStripMenuItem | Выбор языка программы |

Программное средство состоит их 8 модулей, в которых находится код программы. Названия этих модулей программы:

* «AythForm.h» – файл с кодом окна аунтификации;
* «AythForm.cpp» – файл с инициализацией пространства имен для работы программного обеспечения;
* «RegForm.h» – файл с кодом окна регистрации;
* «RegForm.cpp» – файл с инициализацией пространства имен для работы программного обеспечения;
* «MainForm.h» – файл с кодом главного окна;
* «MainForm.cpp» – файл с инициализацией пространства имен для работы программного обеспечения;
* «HelpForm.h» – файл с кодом справочного окна;
* «HelpForm.cpp» – файл с инициализацией пространства имен для работы программного обеспечения.
  1. Функции и элементы управления

В данном разделе будут рассмотрены функции, которые используются в разработке данного программного средства. Для работы в средстве имеются 2 типа функции. Функции которое используются обработчиком событие и функции, которые вызывается командами. Все данные функции будут изображены в приложении А.

* 1. Проектирование справочной системы

В данном программной средстве справочная система, представленная в виде оконной формы с текстом информации о программе и ее пользовании, изображенной на рисунке 5.

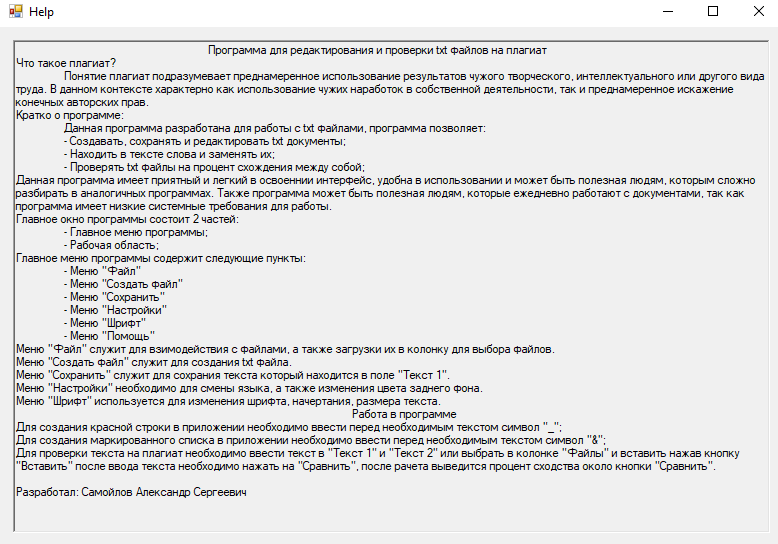


Рисунок 5 – Окно справки

В данной справочной системе имеется один раздел, в котором описывается программа, как правильно взаимодействовать с программой. Данная справочная системы вызывается путем нажатия на кнопку «Помощь», которая расположенная в главном окне программы.

1. Описание программного средства
   1. Общие сведения

При выполнении курсового проекта были использованы следующие библиотеки:

* «fstream» – библиотека предоставляет функционал работы с потоками файлов;
* «iostream» – библиотека предоставляет возможность организации ввода-вывода данных на разных языках;
* «string.h» – библиотека предоставляет возможность работы с строками;
* «stdexcept» – библиотека предоставляет работать с разными типами исключений;
* «Windows.h» – библиотека предоставляющая возможность работы с Windows API.
  1. Входные и выходные данные

В качестве входных данных в программе используются файлы с расширением .txt, которые используются пользователем для редактирования и сохранения данных. Программа имеет возможность создавать и сохранять файлы.

Дополнительно в данном программном средстве используются данные пользователей: логины и пароли.

1. Методика испытаний
   1. Технические требования

Для испытания данного программного средства требуется следующие характеристики:

* процессор частоты работы от 2 ГГц, архитектуры x64;
* оперативная память от 1 ГБ;
* место на диске от 512 МБ;
* клавиатура проводная;
* мышь проводная;
* видеопамять от 64 МБ.

Так же для проведения тестирования было использовано Microsoft Visual Studio 2019.

* 1. Функциональное тестирование

Функциональное тестирование – это тестирование ПО в целях проверки реализации функциональных требований в определенных условиях решать задачи, поставленные пользователем.

В данное программное средстве реализованная система ситуаций и ошибок, которые обрабатывается в ПС, информация о ошибке выводится в выпадающем окне. Пример данного окна представлен на рисунке 6.

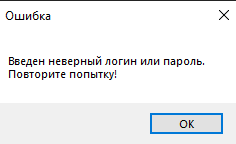


Рисунок 6 – Результат обработки исключений

Таблица 2 – Тест-кейсы для функций

|  |  |  |  |
| --- | --- | --- | --- |
| № | Модуль/функция | Шаги воспроизведения | Результат |
| 1 | Запустить приложение «AntiplagiarismFix» | Открыть директорию приложения. Запустить двойным щелчком файл «AntiplagiarismFix.exe». | Ожидаемый: отображается формы регистрации. |
| Фактический:  фактический результат соответствует ожидаемому, все условия соблюдены. Результаты тестирования приведены на Рисунке Б.1 |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| № | Модуль/функция | Шаги воспроизведения | Результат |
| 2 | Авторизоваться в приложении | Выбрать кнопку «Подтвердить» на форме регистрации. Предварительно корректно заполнить поля «Логин» и «Пароль». Нажать кнопку «Подтвердить». | Ожидаемый: отображается основной формы приложения. |
| Фактический: фактический результат соответствует ожидаемому, все условия соблюдены.  Результаты тестирования приведены на Рисунке Б.2 |
| 3 | Открыть файл | Нажать вкладку меню «Файл». Выбрать в выпадающем списке пункт «Открыть файл». В диалоговом окне выбрать файл. Нажать в диалоговом окне кнопку «Открыть». Затем выбрать нужный файлы в столбце «Файлы» | Ожидаемый: отображение содержимого файла в области содержимого файла. |
| Фактический: фактический результат соответствует ожидаемому, все условия соблюдены. Для примера был использован случайный файл.  Результаты тестирования приведены на Рисунке Б.3 |
| 4 | Сохранение файла | Нажать вкладку меню «Сохранить». Выбрать в выпадающем меню файл, в который вы хотите сохранить файл. | Ожидаемый: сохранения нового содержимого файла. |
| Фактический: фактический результат соответствует ожидаемому, все условия соблюдены.  Результаты тестирования приведены на Рисунке Б.4 |

1. Применение
   1. Назначение программы

Программа разработана для облегчения редактирования текстовых файлов, а также их проверки их на процент сравнения функцией антиплагиата.

Сетевая поддержка не подразумевается из-за ненадобности.

* 1. Условия применения

Для корректной работы приложения требуется соблюдать минимальные системные требования программы, а именно:

* процессор частоты работы от 2 ГГц, архитектуры x64;
* оперативная память от 1 ГБ;
* место на диске от 512 МБ;
* клавиатура проводная;
* мышь проводная;
* видеопамять от 64 МБ;
* операционная система Microsoft Windows 10.

Для работы с данным программным средством необходимо установить Microsoft .Net Framework 4.8. Данный компонент будет предложено установить в ходе установки программы.

* 1. Справочная система

Справочная система реализована с помощь 3 текстовых файлов, которые выводятся в окно помощи. Каждый текстовый файл имеет информацию о приложении, о том, как с ним взаимодействовать и так далее. Данные текстовые файлы локализованы на 3 языка: русский белорусский, английский.

Справочная система разработана в блокноте и является текстовым файлом.

Для вызова справочной системы используется ToolStripItem, который находит на главной окне программного средства. Просмотр справки происходит в новом окне, которое открывается поверх главного окна.

Заключение

Было разработано и создано программное средство, реализующее программу для реализации средства для обработки текстовых файлов с функцией антиплагиата.

Для разработки программного средства было изучено и сформировано знание о WindowsForm, а также знание об основах разработки приложений на С++. В ходе тестирования приходилось переписывать алгоритм для проверки на плагиат, для более глубокой проверки кода.

В ходе разработки данного программного средства было создано множество методов и функций для обработки информации. Были изучены основы Microsoft .Net Framework 4.8 в связке с Windows Form и Windows API.

В данном программе реализованы такие задачи как: авторизация в системе, локализация приложения на 3 языка, изменение заднего фона программы, ввод пользователем текстовой информации, возможность редактирования текстовых файлов, возможность открытия и сохранения текстовых файлов, возможность поиска и замены текста.

Преимуществом данного программного средства является легкость в освоении и низкие системны требования.

Недостатками данного ПС является такие свойства как: ограниченный функционал, отсутствие кроссплатформенности, отсутствие обновлении.

Разработанное программное средство является готовым продуктом, готовым к использованию.

Список информационных источников

1. Багласова, Т.Г. Методические указания по оформлению курсовых и дипломных проектов / Т.Г. Багласова, К.О. Якимович. – Минск: КБП, 2013. – 29 c.
2. Багласова, Т.Г. Методические указания по оформлению курсовых работ, дипломных проектов и отчётов для учащихся специальности 2-40 01 01 «ПОИТ» / Т.Г. Багласова. – Минск: КБП, 2016. – 30 с.
3. Дейтел, Х. Как программировать на C++ / Х. Дейтел, П. Дейтел. – М.: ДМК Пресс, 2018. – 1021 с.
4. Каталог API (Microsoft) и справочных материалов [Электронный ресурс].  – Microsoft, 2020.  –  Режим доступа: http://msdn.microsoft.com/ru-ru/library. – Дата доступа: 20.06.2022
5. Мейерс, С. Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ / С. Мейерс. – М.: ДМК Пресс, 2006. – 300 с.
6. Объектно-ориентированный анализ и проектирование с примерами приложений / Гради Буч [и др.]. –3-е изд. – М.: ООО «И.Д. Вильямс», 2008. – 720с.
7. Общие требования к тестовым документам: ГОСТ 2.105-95. – Введ. 01.01.1996. Минск: Межгос. совет по стандартизации, метрологии и сертификации, 1995. – 84 с.
8. Описание программы. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.402-2000. – Введ. 01.09.2001. – Минск: Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 14 с.

9. Текст программы. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.401-2000. – Введ. 01.09.2001. – Минск: Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 16 с

Приложение А

(обязательное)

Текст программных модулей

#include "AuthForm.h"//Подключение заголовочного файлы AuthForm.h

#include "MainForm.h"//Подключение заголовочного файлы MainForm.h

#include "RegForm.h"//Подключение заголовочного файлы RegForm.h

#include "HelpForm.h"//Подключение заголовочного файлы HelpForm.h

#include <Windows.h>//Подключение библиотеки Windows.h для работы оконных форм

using namespace CourseWork;//Использование пространства имен CourseWork

[STAThreadAttribute]//Указанание атрибута потока однопоточного контейнера

int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)//Вызок функции для запуска оконного приложения, которая принимает 4 параметра и возвращает один целочисленный

{

Application::EnableVisualStyles();//Cтатический метод статистического класса устанавливает доступность визуальный стилей

Application::SetCompatibleTextRenderingDefault(false);//Cтатистический метод статистического класса устанавливает единообразное отображение шрифтов текста

Application::Run(gcnew AuthForm);//Вызов метода Run для запуска оконной формы AuthForm

return 0;//Ворзат из функции значения 0

}

#pragma once

#include <string.h>//подключение библиотеки string.h для работы с строками

#include <fstream>//подключение библиотеки fstream дляx чтения файлов

#include "MainForm.h"//подключение оконной формы MainForm.h

#include "RegForm.h"//подключение оконной формы RegForm.h

namespace CourseWork {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::IO;

/// <summary>

/// Сводка для AuthForm

/// </summary>

public ref class AuthForm : public System::Windows::Forms::Form

{

public:

AuthForm(void)

{

InitializeComponent();

#pragma endregion

int Language = 2;//декланируется и инициализируется глобальная переменная типа int для смены языка в программе

int Color = 0;//декланируется и инициализируется глобальная переменная типа int для смены цвета в программе

private: System::Void blueToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на blueToolStripMenuItem , в котором происходит измение цвета заднего фона на синий цвет

{

this->BackColor = Color::Blue;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Синий"

this->label1->ForeColor=Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

Color = 1;//присвоение значения в Color

}

private: System::Void greenToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на greenToolStripMenuItem , в котором происходит измение цвета заднего фона на зеленый цвет

{

this->BackColor = Color::Green;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Зеленый"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

Color = 2;//присвоение значения в Color

}

private: System::Void redToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на redToolStripMenuItem , в котором происходит измение цвета заднего фона на красный цвет

{

this->BackColor = Color::Red;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Красный"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

Color = 3;//присвоение значения в Color

}

private: System::Void defaultToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на defaultToolStripMenuItem , в котором происходит измение цвета заднего фона стандартный цвет

{

this->BackColor = SystemColors::ButtonFace;//назначить фоновым цветом ЭТОЙ оконной формы ситемный цвет "кнопочный серый"

this->label1->ForeColor = Color::Black;//назначить цвет надписи Label1 "Черный цвет"

this->label2->ForeColor = Color::Black;//назначить цвет надписи Label2 "Черный цвет"

Color = 0;//присвоение значения в Color

}

private: System::Void bYToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) //обработчик нажатия на bYToolStripMenuItem , в котором происходит измение языка на белорусский

{

this->settingsToolStripMenuItem->Text = L"Наладжванне";//заполение текстом меню settingsToolStripMenuItem на белорусском языке

this->languageToolStripMenuItem->Text = L"Мова";//заполение текстом меню languageToolStripMenuItem на белорусском языке

this->rUToolStripMenuItem->Text = L"РУС";//заполение текстом меню rUToolStripMenuItem на белорусском языке

this->bYToolStripMenuItem->Text = L"БЕЛ";//заполение текстом меню bYToolStripMenuItem на белорусском языке

this->eNToolStripMenuItem->Text = L"АНГЛ";//заполение текстом меню eNToolStripMenuItem на белорусском языке

this->backgroundToolStripMenuItem->Text = L"Заднi фон";//заполение текстом меню backgroundToolStripMenuItem на белорусском языке

this->greenToolStripMenuItem->Text = L"Зялёны";//заполение текстом меню greenToolStripMenuItem на белорусском языке

this->blueToolStripMenuItem->Text = L"Сiнi";//заполение текстом меню blueToolStripMenuItem на белорусском языке

this->redToolStripMenuItem->Text = L"Чырвоны";//заполение текстом меню redToolStripMenuItem на белорусском языке

this->defaultToolStripMenuItem->Text = L"Стандартны";//заполение текстом меню defaultToolStripMenuItem на белорусском языке

this->label1->Text = "Лагiн";//заполение текстом надписи label1 на белорусском языке

this->label2->Text = "Пароль";//заполение текстом надписи label2 на белорусском языке

this->button1->Text = "Пацвердзiць";//заполение текстом надписи на кнопке button1 на белорусском языке

this->button2->Text = "Зарэгiстравацца";//заполение текстом надписи на кнопке button2 на белорусском языке

Language = 1;//присвоение значений в Language

}

private: System::Void eNToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)// обработчик нажатия на eNToolStripMenuItem, в котором происходит измение языка на англиский

{

this->settingsToolStripMenuItem->Text = L"Settings";//заполение текстом меню settingsToolStripMenuItem на англиском языке

this->languageToolStripMenuItem->Text = L"Language";//заполение текстом меню languageToolStripMenuItem на англиском языке

this->rUToolStripMenuItem->Text = L"RU";//заполение текстом меню rUToolStripMenuItem на англиском языке

this->bYToolStripMenuItem->Text = L"BY";//заполение текстом меню bYToolStripMenuItem на англиском языке

this->eNToolStripMenuItem->Text = L"EN";//заполение текстом меню eNToolStripMenuItem на англиском языке

this->backgroundToolStripMenuItem->Text = L"Background";//заполение текстом меню backgroundToolStripMenuItem на англиском языке

this->greenToolStripMenuItem->Text = L"Green";//заполение текстом меню greenToolStripMenuItem на англиском языке

this->blueToolStripMenuItem->Text = L"Blue";//заполение текстом меню blueToolStripMenuItem на англиском языке

this->redToolStripMenuItem->Text = L"Red";//заполение текстом меню redToolStripMenuItem на англиском языке

this->defaultToolStripMenuItem->Text = L"Default";//заполение текстом меню defaultToolStripMenuItem на англиском языке

this->label1->Text = "Login";//заполение текстом надписи label1 на англиском языке

this->label2->Text = "Password";//заполение текстом надписи label2 на англиском языке

this->button1->Text = "Confirm";//заполение текстом надписи на кнопке button1 на англиском языке

this->button2->Text = "Register";//заполение текстом надписи на кнопке button2 на англиском языке

Language = 2;//присвоение значений в Language

}

private: System::Void rUToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)// обработчик нажатия на rUToolStripMenuItem, в котором происходит измение языка на русский

{

this->settingsToolStripMenuItem->Text = L"Настройки";//заполение текстом меню settingsToolStripMenuItem на русском языке

this->languageToolStripMenuItem->Text = L"Язык";//заполение текстом меню languageToolStripMenuItem на русском языке

this->rUToolStripMenuItem->Text = L"РУС";//заполение текстом меню rUToolStripMenuItem на русском языке

this->bYToolStripMenuItem->Text = L"БЕЛ";//заполение текстом меню bYToolStripMenuItem на русском языке

this->eNToolStripMenuItem->Text = L"АНГЛ";//заполение текстом меню eNToolStripMenuItem на русском языке

this->backgroundToolStripMenuItem->Text = L"Задний фон";//заполение текстом меню backgroundToolStripMenuItem на русском языке

this->greenToolStripMenuItem->Text = L"Зеленый";//заполение текстом меню greenToolStripMenuItem на русском языке

this->blueToolStripMenuItem->Text = L"Синий";//заполение текстом меню blueToolStripMenuItem на русском языке

this->redToolStripMenuItem->Text = L"Красный";//заполение текстом меню redToolStripMenuItem на русском языке

this->defaultToolStripMenuItem->Text = L"Стандартный";//заполение текстом меню defaultToolStripMenuItem на русском языке

this->label1->Text = "Логин";//заполение текстом надписи label1 на русском языке

this->label2->Text = "Пароль";//заполение текстом надписи label2 на русском языке

this->button1->Text = "Подтвердить";//заполение текстом надписи на кнопке button1 на русском языке

this->button2->Text = "Зарегистрироваться";//заполение текстом надписи на кнопке button2 на русском языке

Language = 3;//присвоение значений в Language

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на button1, в котором проходит аунтификация пользователя

{

try//начало области кода, в которой отслеживаются ошибки

{

int pass = 0;//дикларация и инициализация переменной типа int

StreamReader^ sr = gcnew System::IO::StreamReader("Data.txt");//создвние потока sr для чтение Data.txt

String^ Login = this->textBox1->Text;//дикларация и инициализация переменной типа String значения из textBox1

String^ Password = this->textBox2->Text;//дикларация и инициализация переменной типа String значения из textBox2

String^ FullPass = gcnew String(Login + "#" + Password);//декланируется и инициализируется FullPass путем соединения Login и Password

while (!sr->EndOfStream)//выполнять цикл пока не будет достигнут конец потока sw

{

String^ Text = sr->ReadLine();//декланируется и инициализируется Text путем записи текста из строки документа Data.txt

if (Text == FullPass)// если Text = Fullpass

{

pass++;//увеличение значения pass на 1

break;//прекратить цикл

}

}

if (pass == 1)//если pass = 1

{

MainForm^ form = gcnew MainForm(Language, Color);//создать новую оконную форму типа класса MainForm конструктором с целочисленнными параметрами

AuthForm::Hide();//скрыть AuthForm

form->Show();//показать форму MainForm

}

else//иначе

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Уведзены няправiльны лагiн цi пароль. \nПаутарыце спробу!", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("Invalid logs or password entered. \nTry again!", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на русском языке

{

MessageBox::Show("Введен неверный логин или пароль. \nПовторите попытку!", "Ошибка");//показать сообщение с описание ошибки

}

}

}

catch (System::IO::FileNotFoundException ^ e)//отлов ошибки типа FileNotFoundException

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("База пароляў пустая, зарэгіструйцеся!", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("Password database is empty, register!", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на русском языке

{

MessageBox::Show("База паролей пуста, зарегистрируйтесь!", "Ошибка");//показать сообщение с описание ошибки

}

std::ofstream error("Data.txt");//создание txt файла с именем Data.txt

error.close();//закрытие потока error

}

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {//обработчик нажатия на button2, в котором проходит регистрация пользователя

RegForm^ form = gcnew RegForm(Language, Color);//создать новую оконную форму типа класса RegForm конструктором с целочисленными параметрами

form->Show();//показать форму RegForm

}

};

}

#include "MainForm.h"

#pragma once

#include <string.h>//подключение библиотеки string.h для работы с строками

#include <fstream>//подключение библиотеки fstream дляx чтения файлов

#include <stdexcept>//подключение библиотеки stdexcept для работы с исключениями

#include "HelpForm.h"//подключение оконной формы HelpForm.h

namespace CourseWork {

/// <summary>

/// Сводка для MainForm

/// </summary>

public ref class MainForm : public System::Windows::Forms::Form

{

public:

MainForm(int x, int y)//коструктор принимающий несколько целочисленных параметров

{

InitializeComponent();//метод InitializeComponent()

this->Color = y;//инициализируем глобальную переменную Color

this->Language = x;//инициализируем глобальную переменную Color

}

#pragma endregion

void DeleteTNR(String^& x)//функция для удаления из строки \t \n \r

{

bool point = false;//декларация и инициализация переменной типа bool

int poi = 0;//декларация и инициализация переменной типа int

String^ Text = gcnew String(x);//декларация и инициализация переменной типа String

String^ СorrectedText;//декларация переменной типа String

int Length = Text->Length;//декларация и инициализация переменной типа int с записью длины строки Text

for (int i = 0; i < Length; i++)//цикл для нахождения симоволов в строке

{

if ((Text[i] == '\t') | (Text[i] == '\n') | (Text[i] == '\r'))//если в данной строке найдено '\t', '\n', '\r'

{

if (i > 2)//если i > 2

{

if (point != true)//если point не равен true

{

poi = i;//присвоение poi значения i

if (Text[i - 1] != '.')

{

СorrectedText += ". ";//добавление значения ". " в CorrectedText

point = true;//присвоение point значения true

}

else//иначе

{

СorrectedText += " ";//добавление значения " " в CorrectedText

}

}

}

continue;//переход на следющую итеррация в цикле

}

СorrectedText += Text[i];//добавление значения из Text[i] в CorrectedText

if (i - poi > 2)//если i-poi > 2

{

point = false;//присвоение point значения false

}

if (Length == i)//если Length = i

{

СorrectedText += ".";//добавление значения "." в CorrectedText

}

}

DelSpace(x = СorrectedText);//вызов функции DelSpace

}

void DelSpace(String^& x)//функция для удаления двойных пробелов подряд в строке

{

String^ Text = gcnew String(x);//декларация и инициализация переменной типа String

String^ СorrectedText;//декларация переменной типа String

int Length = Text->Length;//декларация и инициализация переменной типа int с записью длины строки Text

for (int i = 0; i < Length; i++)//цикл для нахождения симоволов в строке

{

if (i != Length - 1)//если i не равен Length - 1

{

if (Text[i] == ' ' & Text[i + 1] == ' ')//если в строке надены символы ' '

{

continue;//переход на следющую итеррация в цикле

}

}

if (i + 1 == Length)//если i + 1 равен Length

{

if (Text[i] == ' ')//если значение из Text[i] равно ' '

{

continue;//переход на следющую итеррация в цикле

}

}

СorrectedText += Text[i];//добавление символа Text[i] в СorrectedText

if (i + 1 == Length)//если i+1 = Lenght

{

if (Text[i] != '.' & Text[i] != ' ')//если в строке не надены символы '.' и ' '

{

СorrectedText += ".";//добавление символа "." в СorrectedText

}

}

}

DeleteMark (x = СorrectedText);//вызов функции DeleteMark

}

void DeleteMark(String^& x)//функция для удаления знаков препиная в строке таких как '-',';','"','\','!','?','.'

{

String^ Text = gcnew String(x);//декларация и инициализация переменной типа String

String^ СorrectedText;//декларация переменной типа String

int Length = Text->Length;//декларация и инициализация переменной типа int с записью длины строки Text

for (int i = 0; i < Length; i++)//цикл для нахождения симоволов в строке

{

if (Text[i] == '.' | Text[i] == '?' | Text[i] == '!')//если в строке обнаруженны данные символы '.','?','!'

{

СorrectedText += ".";//добавление символа "." в СorrectedText

}

else//иначе

{

if (Text[i] == '-' | Text[i] == ';' | Text[i] == ':' | Text[i] == '"' | Text[i] == '\'')//если в строке обнаруженны данные символы '-',';',':','"','\''

{

continue;//переход на следющую итеррация в цикле

}

else//иначе

{

СorrectedText += Text[i];//добавление символа Text[i] в СorrectedText

}

}

}

DeleteDoublePoint(x = СorrectedText);//вызов функции DeleteDoublePoint

}

void DeleteDoublePoint(String^& x)//функция для удаления двойных точек стоящих рядом в строке

{

String^ Text = gcnew String(x);//декларация и инициализация переменной типа String

String^ СorrectedText;//декларация переменной типа String

int Length = Text->Length;//декларация и инициализация переменной типа int с записью длины строки Text

for (int i = 0; i < Length; i++)//цикл для нахождения симоволов в строке

{

if (i != 0)//если i не равен 0

{

if (Text[i] == '.' & Text[i - 1] == '.')

{

continue;//переход на следющую итеррация в цикле

}

else//иначе

{

СorrectedText += Text[i];//добавление символа Text[i] в СorrectedText

}

}

else//иначе

{

СorrectedText += Text[i];//добавление символа Text[i] в СorrectedText

}

}

DeleteFirstPoint(x = СorrectedText);//вызов функции DeleteFirstPoint

}

void DeleteFirstPoint(String^& x)//функция для удаления точки если строка начинается с точки

{

String^ Text = gcnew String(x);//декларация и инициализация переменной типа String

String^ СorrectedText;//декларация переменной типа String

int Length = Text->Length;//декларация и инициализация переменной типа int с записью длины строки Text

for (int i = 0; i < Length; i++)//цикл для нахождения симоволов в строке

{

if (i == 0)//если i = 0

{

if (Text[i] == '.' & Text[i+1] == ' ')//если в строке найдены символы '. '

{

continue;//переход на следющую итеррация в цикле

}

}

else//иначе

{

СorrectedText += Text[i];//добавление символа Text[i] в СorrectedText

}

}

x = СorrectedText;//присвоение значения CorrectedText в строку x

}

void DeleteDoubleSentences(String^& x)//функция для удаления двойных пробелов в строке

{

String^ Text = gcnew String(x);//декларация и инициализация переменной типа String

String^ СorrectedText;//декларация переменной типа String

int Length = Text->Length;//декларация и инициализация переменной типа int с записью длины строки Text

for (int i = 0; i < Length; i++)//цикл для нахождения симоволов в строке

{

if (i != Length - 1 & i != 0)//если i не равен Lenght-1 и i не равен 0

{

if ((Text[i] == ' ' & Text[i + 1] == '.') | (Text[i - 1] == ' ' & Text[i] == '.'))//если в строке обнарежны связки символов

{

i++;//увеличение значения i на 1

continue;//переход на следющую итеррация в цикле

}

else//иначе

{

СorrectedText += Text[i];//добавление символа Text[i] в СorrectedText

}

}

else//иначе

{

СorrectedText += Text[i];//добавление символа Text[i] в СorrectedText

}

}

x = СorrectedText;//присвоение значения CorrectedText в строку x

}

int Language = 0;//декларация и инициализация глобальной переменной типа int для смены языка в программе

int Color = 0;//декларация и инициализация глобальной переменной типа int для смены цвета фона в программе

int LengthText = 0;//декларация и инициализация глобальной переменной типа int

int LengthText2 = 0;//декларация и инициализация глобальной переменной типа int

int pass = 1;//декларация и инициализация глобальной переменной типа int

String^ URLF;//декларация глобальной переменной типа String

String^ Range;//декларация глобальной переменной типа Stirng

private: System::Void MainForm\_Load(System::Object^ sender, System::EventArgs^ e)//метот-обработчик события загрузки оконной формы

{

if (Language == 1)//если Language = 1 => установка белорусского языка

{

this->label1->Text = "Тэкст 1";//заполение текстом надписи label1 на белорусском языке

this->label2->Text = "Тэкст 2";//заполение текстом надписи label2 на белорусском языке

this->label4->Text = "Файлы";//заполение текстом надписи label2 на белорусском языке

this->button1->Text = "Параунаць";//заполение текстом кнопки button1 на белорусском языке

this->button2->Text = "Уставiць сюды";//заполение текстом кнопки button2 на белорусском языке

this->button3->Text = "Уставiць сюды";//заполение текстом кнопки button3 на белорусском языке

this->button4->Text = "Пошук";//заполение текстом кнопки button4 на белорусском языке

this->button5->Text = "Замяніць";//заполение текстом кнопки button5 на белорусском языке

this->fileToolStripMenuItem->Text = "Файл";//заполение текстом меню fileToolStripMenuItem на белорусском языке

this->openFileToolStripMenuItem->Text = "Аткрыць";//заполение текстом меню openFileToolStripMenuItem на белорусском языке

this->saveToolStripMenuItem->Text = "Захаваць";//заполение текстом меню saveToolStripMenuItem на белорусском языке

this->settingsToolStripMenuItem->Text = "Наладжваньне";//заполение текстом меню settingsToolStripMenuItem на белорусском языке

this->backGroundToolStripMenuItem->Text = "Заднi фон";//заполение текстом меню backGroundToolStripMenuItem на белорусском языке

this->languageToolStripMenuItem->Text = "Мова";//заполение текстом меню languageToolStripMenuItem на белорусском языке

this->rUToolStripMenuItem->Text = "РУС";//заполение текстом меню rUToolStripMenuItem на белорусском языке

this->bYToolStripMenuItem->Text = "БЕЛ";//заполение текстом меню bYToolStripMenuItem на белорусском языке

this->eNToolStripMenuItem->Text = "АНГЛ";//заполение текстом меню eNToolStripMenuItem на белорусском языке

this->greenToolStripMenuItem->Text = "Зялёны";//заполение текстом меню greenToolStripMenuItem на белорусском языке

this->blueToolStripMenuItem->Text = "Сiнi";//заполение текстом меню blueToolStripMenuItem на белорусском языке

this->redToolStripMenuItem->Text = "Чырвоны";//заполение текстом меню redToolStripMenuItem на белорусском языке

this->defaultToolStripMenuItem->Text = "Стандартны";//заполение текстом меню defaultToolStripMenuItem на белорусском языке

this->createFileToolStripMenuItem->Text = "Стварьць файл";//заполение текстом меню createFileToolStripMenuItem на белорусском языке

this->fontToolStripMenuItem->Text = "Шрыфт";//заполение текстом меню fontToolStripMenuItem на белорусском языке

this->helpToolStripMenuItem->Text = "Дапамога";//заполение текстом меню helpToolStripMenuItem на белорусском языке

}

if (Language == 2)//если Language = 2 => установка англиского языка

{

this->label1->Text = "Text 1";//заполение текстом надписи label1 на англиском языке

this->label2->Text = "Text 2";//заполение текстом надписи label2 на англиском языке

this->label4->Text = "Files";//заполение текстом надписи label4 на англиском языке

this->button1->Text = "Compare";//заполение текстом кнопки button1 на англиском языке

this->button2->Text = "Paste here";//заполение текстом кнопки button2 на англиском языке

this->button3->Text = "Paste here";//заполение текстом кнопки button3 на англиском языке

this->button4->Text = "Search";//заполение текстом кнопки button4 на англиском языке

this->button5->Text = "Replace";//заполение текстом кнопки button5 на англиском языке

this->backGroundToolStripMenuItem->Text = "Backgroud";//заполение текстом меню backGroundToolStripMenuItem на белорусском языке

this->fileToolStripMenuItem->Text = L"File";//заполение текстом меню fileToolStripMenuItem на англиском языке

this->openFileToolStripMenuItem->Text = L"Open";//заполение текстом меню openFileToolStripMenuItem на англиском языке

this->saveToolStripMenuItem->Text = "Save";//заполение текстом меню saveToolStripMenuItem на англиском языке

this->settingsToolStripMenuItem->Text = "Settings";//заполение текстом меню settingsToolStripMenuItem на англиском языке

this->rUToolStripMenuItem->Text ="RU";//заполение текстом меню rUToolStripMenuItem на англиском языке

this->bYToolStripMenuItem->Text = "BY";//заполение текстом меню bYToolStripMenuItem на англиском языке

this->eNToolStripMenuItem->Text = "EN";//заполение текстом меню eNToolStripMenuItem на англиском языке

this->greenToolStripMenuItem->Text = "Green";//заполение текстом меню greenToolStripMenuItem на англиском языке

this->blueToolStripMenuItem->Text = "Blue";//заполение текстом меню blueToolStripMenuItem на англиском языке

this->redToolStripMenuItem->Text = "Red";//заполение текстом меню redToolStripMenuItem на англиском языке

this->defaultToolStripMenuItem->Text = "Default";//заполение текстом меню defaultToolStripMenuItem на англиском языке

this->createFileToolStripMenuItem->Text = "Create file";//заполение текстом меню createFileToolStripMenuItem на англиском языке

this->languageToolStripMenuItem->Text = "Language";//заполение текстом меню languageToolStripMenuItem на англиском языке

this->fontToolStripMenuItem->Text = "Font";//заполение текстом меню fontToolStripMenuItem на англиском языке

this->helpToolStripMenuItem->Text = "Help";//заполение текстом меню helpToolStripMenuItem на англиском языке

}

if (Language == 3)//если Language = 3 => установка русского языка

{

this->label1->Text = "Tекст 1";//заполение текстом надписи label1 на русском языке

this->label2->Text = "Текст 2";//заполение текстом надписи label2 на русском языке

this->label4->Text = "Файлы";//заполение текстом надписи label4 на русском языке

this->button1->Text = "Сравнить";//заполение текстом кнопки button1 на русском языке

this->button2->Text = "Вставить сюда";//заполение текстом кнопки button2 на русском языке

this->button3->Text = "Вставить сюда";//заполение текстом кнопки button3 на русском языке

this->button4->Text = "Поиск";//заполение текстом кнопки button4 на русском языке

this->button5->Text = "Заменить";//заполение текстом кнопки button5 на русском языке

this->fileToolStripMenuItem->Text = L"Файл";//заполение текстом меню fileToolStripMenuItem на англиском языке

this->openFileToolStripMenuItem->Text = L"Открыть";//заполение текстом меню openFileToolStripMenuItem на англиском языке

this->saveToolStripMenuItem->Text = "Сохранить";//заполение текстом меню saveToolStripMenuItem на англиском языке

this->backGroundToolStripMenuItem->Text = "Задний фон";//заполение текстом меню backGroundToolStripMenuItem на англиском языке

this->settingsToolStripMenuItem->Text = "Настройки";//заполение текстом меню settingsToolStripMenuItem на англиском языке

this->rUToolStripMenuItem->Text = L"РУС";//заполение текстом меню rUToolStripMenuItem на англиском языке

this->bYToolStripMenuItem->Text = L"БЕЛ";//заполение текстом меню bYToolStripMenuItem на англиском языке

this->eNToolStripMenuItem->Text = L"АНГЛ";//заполение текстом меню eNToolStripMenuItem на англиском языке

this->greenToolStripMenuItem->Text = L"Зеленый";//заполение текстом меню greenToolStripMenuItem на англиском языке

this->blueToolStripMenuItem->Text = L"Синий";//заполение текстом меню blueToolStripMenuItem на англиском языке

this->redToolStripMenuItem->Text = L"Красный";//заполение текстом меню redToolStripMenuItem на англиском языке

this->defaultToolStripMenuItem->Text = L"Стандартный";//заполение текстом меню defaultToolStripMenuItem на англиском языке

this->createFileToolStripMenuItem->Text = "Создать файл";//заполение текстом меню createFileToolStripMenuItem на англиском языке

this->languageToolStripMenuItem->Text = "Язык";//заполение текстом меню languageToolStripMenuItem на англиском языке

this->fontToolStripMenuItem->Text = "Шрифт";//заполение текстом меню fontToolStripMenuItem на англиском языке

this->helpToolStripMenuItem->Text = "Помощь";//заполение текстом меню helpToolStripMenuItem на англиском языке

}

if (Color == 0)//если Color = 0 => установка фонового цвета "Кнопочный серый"

{

this->BackColor = SystemColors::ButtonFace;//назначить фоновым цветом ЭТОЙ оконной формы ситемный цвет "кнопочный серый"

this->label1->ForeColor = Color::Black;//назначить цвет надписи Label1 "Черный цвет"

this->label2->ForeColor = Color::Black;//назначить цвет надписи Label2 "Черный цвет"

this->label4->ForeColor = Color::Black;//назначить цвет надписи Label4 "Черный цвет"

}

if (Color == 1)//если Color = 1 => установка фонового цвета "Синий"

{

this->BackColor = Color::Blue;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Синий"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

this->label4->ForeColor = Color::White;//назначить цвет надписи Label4 "Белый"

}

if (Color == 2)//если Color = 2 => установка фонового цвета "Зеленый"

{

this->BackColor = Color::Green;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Зеленый"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

this->label4->ForeColor = Color::White;//назначить цвет надписи Label4 "Белый"

}

if (Color == 3)//если Color = 3 => установка фонового цвета "Красный"

{

this->BackColor = Color::Red;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Красный"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

this->label4->ForeColor = Color::White;//назначить цвет надписи Label4 "Белый"

}

}

private: System::Void openFileToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на saveToolStripMenuItem, в котором происходит загрузка файлов в fileListBox1

{

fileListBox1->Pattern = "\*.txt";//фильтация типов файлов в fileListBox1

folderBrowserDialog1->ShowDialog();//открыть окно просмотра папок

fileListBox1->FileName = folderBrowserDialog1->SelectedPath;//отображение файлов только из папки выбраноой в folderBrowserDialog1

}

private: System::Void saveToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на saveToolStripMenuItem, в котором происходит сохранение текста из richTextBox1 в файл

{

try//начало области кода, в которой отслеживаются ошибки

{

String^ Text = this->richTextBox1->Text;//декларация и инициализация переменной типа String текстом из richTextBox1

if (Text->Length == 0)//если строка пустая

{

throw 1;//вызов исключения типа int

}

openFileDialog1->Filter = "Text Files|\*.txt";//фильтрация файлов в openFileDialog1

if (openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK)

{

StreamWriter^ sw = gcnew StreamWriter(openFileDialog1->FileName);//создание потока для записи информации в выбранное место

String^ SaveText;//декларация переменной типа String

String^ proposal;//декларация переменной типа String

int Length = 0;//декларация и инициализация переменной типа String

Length = Text->Length;//присвоение lenght значения длина строки Text

for (int i = 0; i < Length; i++)//поиск символа в строке

{

if (Text[i] == '\_')//красная строка

{

SaveText += "\n\t";//добаление перехода на новую строку и табулирования

}

else//иначе

{

if (Text[i] == '&')//маркированный список

{

SaveText += "\n\t- ";//добаление перехода на новую строку, табулирования и символа "-"

}

else//иначе

{

SaveText += Text[i];//добавление символа Text[i] в SaveText

}

}

}

sw->WriteLine(SaveText);//запись строки в Data.txt

sw->Close();//закрытие потока sw

if (Language == 1)//если Language = 1 => MessageBox белорусском языке

{

MessageBox::Show("Файл захаваны в : " + openFileDialog1->FileName, "Захаваны");//показать сообщение с информацией

}

if (Language == 2)//если Language = 1 => MessageBox англиском языка

{

MessageBox::Show("File save in : " + openFileDialog1->FileName, "Save");//показать сообщение с информацией

}

if (Language == 3)//если Language = 1 => MessageBox русском языка

{

MessageBox::Show("Файл сохраненен в " +openFileDialog1->FileName, "Сохранено");//показать сообщение с информацией

}

}

}

catch (int)//отлов ошибки типа int

{

if (Language == 1)//если Language = 1 => MessageBox белорусском языке

{

MessageBox::Show("Текстовое поле пустое, увядзiце што-небудзь", "Памылка");//показать сообщение с информацией

}

if (Language == 2)//если Language = 1 => MessageBox англиском языка

{

MessageBox::Show("Text field is empty, enter something", "Error");//показать сообщение с информацией

}

if (Language == 3)//если Language = 1 => MessageBox русском языка

{

MessageBox::Show("Тектвовое поле пустое, введите что-нибудь", "Ошибка");//показать сообщение с информацией

}

}

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на button1, в котором происходит выполнение процесса сравнения текста из richTextBox1 и richTextBox2

{

try//начало области кода, в которой отслеживаются ошибки

{

String^ Sourse = richTextBox1->Text;//декларация и инициализация текстом из richTextBox1 переменной типа String

String^ paste = richTextBox2->Text;//декларация и инициализация текстом из richTextBox2 переменной типа String

if (Sourse->Length == 0 | paste->Length == 0)//если строки пустые

{

throw 1;//вызов исключения типа int

}

DeleteTNR(Sourse);//вызов функции DeleteTNR для строки Sourse

DeleteTNR(paste);//вызов функции DeleteTNR для строки paste

DelSpace(Sourse);//вызов функции DelSpace для строки Sourse

DelSpace(paste);//вызов функции DelSpace для строки paste

DeleteDoubleSentences(Sourse);//вызов функции DeleteDoubleSentences для строки Sourse

DeleteDoubleSentences(paste);//вызов функции DeleteDoubleSentences для строки paste

double type;//декларация переменной типа double

int fp = 0;//декларация с инициализацией переменной типа int

int fp2 = 0;//декларация с инициализацией переменной типа int

int sentence = 0;//декларация с инициализацией переменной типа int

int sp, sp2;//декларация переменной типа int

int sim = 0;//декларация с инициализацией переменной типа int

int Length = Sourse->Length;//декларация переменной с присвоение длины строки Sourse

int Length2 = paste->Length;//декларация переменной с присвоение длины строки paste

for (int i = 0; i <= Length; i++)//цикл для нахождения симоволов в строке

{

if (i == Length)//если i = Lenght

{

break;//выход из цикла

}

if (Sourse[i] == '.' | Sourse[i] == '?' | Sourse[i] == '!')//если в строке встретились данные символы '.','?','!'

{

sentence++;//увеличение счетчика предложений

sp = i + 1;//переменная значение которой индекс, которой укакзывает на расположение точки в строке

String^ sent;//декларация переменной типа String

for (int j = fp; j < sp; j++)//цикл для записи предложения в строку

{

sent += Sourse[j];//добавление символа Sourse[j] в sent

}

for (int w = 0; w < Length2; w++)//цикл для нахождения симоволов в строке

{

if (paste[w] == '.' | paste[w] == '?' | paste[w] == '!')//если в строке найдены символы '.','?','!'

{

sp2 = w + 1;//переменная значение которой индекс, которой укакзывает на расположение точки в строке

String^ sent2;//декларация переменной типа String

for (int e = fp2; e < sp2; e++)//цикл для записи предложения в строку

{

sent2 += paste[e];//добавление символа paste[e] в sent2

}

sent = sent->ToLower();//уменьшение регистра строки sent

sent2 = sent2->ToLower();//уменьшение регистра строки sent 2

if (sent == sent2)//если sent = sent

{

sim += 1;//увеличение счетчика похожих предложений

}

fp2 = sp2 + 1;//присвоение значений переменной fp2 значений sp2 + 1

}

}

fp = sp + 1;//присвоение значений переменной fp значений sp + 1

fp2 = 0;//обнуление переменной

}

}

if (sentence == 0)//если кол-во предложений равно 0

{

throw 0.1;//вызов исключения типа double

}

double x = double(sim) / double(sentence) \* 100;//вычисление процента сходства текстов x = sim/sentence\*100

if (x > 100)//если x > 100

{

x = 100;//присвоение значений переменной x

}

if (x > 50)//если x < 50

{

this->label3->ForeColor = Color::Red;//изменение цвета шрифта на красный в label3

this->label3->Text = x.ToString() + "%";//присвоение значенией

}

else//иначе

{

this->label3->ForeColor = Color::Green;//изменение цвета шрифта на зеленый в label3

this->label3->Text = x.ToString() + "%";//изменение цвета шрифта в label3

}

}

catch (int)//отлов ошибки типа int

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Текстовое поле пустое, увядзiце што-небудзь","Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("Text field is empty, enter something", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на руссоком языке

{

MessageBox::Show("Тектвовое поле пустое, введите что-нибудь","Ошибка");//показать сообщение с описание ошибки

}

}

catch (double)//отлов ошибки типа double

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Тэкставае поле не мае цэлай прапановы", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("Text field does not have a whole sentence", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на руссоком языке

{

MessageBox::Show("Текстовое поле не имеет целого предложения", "Ошибка");//показать сообщение с описание ошибки

}

}

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на button2, в котором происходит перенос текста из txt файла в richTextBox1

{

try//начало области кода, в которой отслеживаются ошибки

{

StreamReader^ read = gcnew StreamReader(URLF);//создание потока read для чтения для файла, путь которого расположен в сроке URLF

richTextBox1->Text = read->ReadToEnd();//присвоение значений в richTextBox2 из файла

String^ Text = richTextBox1->Text;//декларация и инициализация переменной типа String текстом из richTextBox1

String^ Sentensens;//декларация переменной типа String

read->Close();//закрытие потока read

int counter = 0;//декларация и инициализация переменной типа int

int counterText = 0;//декларация и инициализация переменной типа int

int f = 0;//декларация и инициализация переменной типа int

int l = 0;//декларация и инициализация переменной типа int

int Length = Text->Length;//декларация и инициализация переменной типа int значением длины Text

for (int i = 0; i < Length; i++)//цикл для нахождения симоволов в строке

{

if (Text[i] == '.')//если значение из Text[i] равен '.'

{

counter++;//увеличение счетчика на 1

l = i;//присвоение значений из i в l

if (counter == 1)//если счетчик равен 1

{

for (int w = 0; w < l + 1; w++)//цикл для заполнения строки

{

Sentensens += Text[w];//добавление символа Text[w] в Sentensens

}

}

}

}

LengthText = Length;//присвоение значений Length в LengthText

}

catch (System::ArgumentNullException^ e)//отлов ошибки типа ArgumentNullException

{

if (Language == 1)//показать сообщение с описание ошибки

{

MessageBox::Show("Вы не выбралi файл для адкрыцця", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("You have not selected file to open", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на руссоком языке

{

MessageBox::Show("Вы не выбрали файл для открытия", "Ошибка");//показать сообщение с описание ошибки

}

}

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на button3, в котором происходит перенос текста из txt файла в richTextBox1

{

try//начало области кода, в которой отслеживаются ошибки

{

StreamReader^ read = gcnew StreamReader(URLF);//создание потока read для чтения для файла, путь которого расположен в сроке URLF

richTextBox2->Text = read->ReadToEnd();//присвоение значений в richTextBox2 из файла

read->Close();//закрытие потока read

String^ Text = richTextBox2->Text;//декларация и инициализация переменной типа String текстом из richTextBox2

}

catch (System::ArgumentNullException^ e)//отлов ошибки типа ArgumentNullException

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Вы не выбралi файл для адкрыцця", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("You have not selected file to open", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на руссоком языке

{

MessageBox::Show("Вы не выбрали файл для открытия", "Ошибка");//показать сообщение с описание ошибки

}

}

}

private: System::Void blueToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на blueToolStripMenuItem , в котором происходит измение цвета заднего фона на синий цвет

{

this->BackColor = Color::Blue;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Синий"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

Color = 1;//присвоение значений в Color

}

private: System::Void greenToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на greenToolStripMenuItem , в котором происходит измение цвета заднего фона на зеленый цвет

{

this->BackColor = Color::Green;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Зеленый"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

Color = 2;//присвоение значений в Color

}

private: System::Void redToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на redToolStripMenuItem , в котором происходит измение цвета заднего фона на красный цвет

{

this->BackColor = Color::Red;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Красный"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

Color = 3;//присвоение значений в Color

}

private: System::Void defaultToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на defaultToolStripMenuItem , в котором происходит измение цвета заднего фона стандартный цвет

{

this->BackColor = SystemColors::ButtonFace;//назначить фоновым цветом ЭТОЙ оконной формы ситемный цвет "кнопочный серый"

this->label1->ForeColor = Color::Black;//назначить цвет надписи Label1 "Черный цвет"

this->label2->ForeColor = Color::Black;//назначить цвет надписи Label2 "Черный цвет"

Color = 0;//присвоение значений в Color

}

private: System::Void bYToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) //обработчик нажатия на bYToolStripMenuItem , в котором происходит измение языка на белорусский

{

Language = 1;//присвоение значений в Language

this->label1->Text = "Тэкст 1";//заполение текстом надписи label1 на белорусском языке

this->label2->Text = "Тэкст 2";//заполение текстом надписи label2 на белорусском языке

this->label4->Text = "Файлы";//заполение текстом надписи label2 на белорусском языке

this->button1->Text = "Параунаць";//заполение текстом кнопки button1 на белорусском языке

this->button2->Text = "Уставiць сюды";//заполение текстом кнопки button2 на белорусском языке

this->button3->Text = "Уставiць сюды";//заполение текстом кнопки button3 на белорусском языке

this->button4->Text = "Пошук";//заполение текстом кнопки button4 на белорусском языке

this->button5->Text = "Замяніць";//заполение текстом кнопки button5 на белорусском языке

this->fileToolStripMenuItem->Text = "Файл";//заполение текстом меню fileToolStripMenuItem на белорусском языке

this->openFileToolStripMenuItem->Text = "Аткрыць";//заполение текстом меню openFileToolStripMenuItem на белорусском языке

this->saveToolStripMenuItem->Text = "Захаваць";//заполение текстом меню saveToolStripMenuItem на белорусском языке

this->settingsToolStripMenuItem->Text = "Наладжваньне";//заполение текстом меню settingsToolStripMenuItem на белорусском языке

this->backGroundToolStripMenuItem->Text = "Заднi фон";//заполение текстом меню backGroundToolStripMenuItem на белорусском языке

this->languageToolStripMenuItem->Text = "Мова";//заполение текстом меню languageToolStripMenuItem на белорусском языке

this->rUToolStripMenuItem->Text = "РУС";//заполение текстом меню rUToolStripMenuItem на белорусском языке

this->bYToolStripMenuItem->Text = "БЕЛ";//заполение текстом меню bYToolStripMenuItem на белорусском языке

this->eNToolStripMenuItem->Text = "АНГЛ";//заполение текстом меню eNToolStripMenuItem на белорусском языке

this->greenToolStripMenuItem->Text = "Зялёны";//заполение текстом меню greenToolStripMenuItem на белорусском языке

this->blueToolStripMenuItem->Text = "Сiнi";//заполение текстом меню blueToolStripMenuItem на белорусском языке

this->redToolStripMenuItem->Text = "Чырвоны";//заполение текстом меню redToolStripMenuItem на белорусском языке

this->defaultToolStripMenuItem->Text = "Стандартны";//заполение текстом меню defaultToolStripMenuItem на белорусском языке

this->createFileToolStripMenuItem->Text = "Стварьць файл";//заполение текстом меню createFileToolStripMenuItem на белорусском языке

this->fontToolStripMenuItem->Text = "Шрыфт";//заполение текстом меню fontToolStripMenuItem на белорусском языке

this->helpToolStripMenuItem->Text = "Дапамога";//заполение текстом меню helpToolStripMenuItem на белорусском языке

}

private: System::Void eNToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) // обработчик нажатия на eNToolStripMenuItem, в котором происходит измение языка на англиский

{

Language = 2;//присвоение значений в Language

this->label1->Text = "Text 1";//заполение текстом надписи label1 на англиском языке

this->label2->Text = "Text 2";//заполение текстом надписи label2 на англиском языке

this->label4->Text = "Files";//заполение текстом надписи label4 на англиском языке

this->button1->Text = "Compare";//заполение текстом кнопки button1 на англиском языке

this->button2->Text = "Paste here";//заполение текстом кнопки button2 на англиском языке

this->button3->Text = "Paste here";//заполение текстом кнопки button3 на англиском языке

this->button4->Text = "Search";//заполение текстом кнопки button4 на англиском языке

this->button5->Text = "Replace";//заполение текстом кнопки button5 на англиском языке

this->backGroundToolStripMenuItem->Text = "Backgroud";//заполение текстом меню backGroundToolStripMenuItem на белорусском языке

this->fileToolStripMenuItem->Text = L"File";//заполение текстом меню fileToolStripMenuItem на англиском языке

this->openFileToolStripMenuItem->Text = L"Open";//заполение текстом меню openFileToolStripMenuItem на англиском языке

this->saveToolStripMenuItem->Text = "Save";//заполение текстом меню saveToolStripMenuItem на англиском языке

this->settingsToolStripMenuItem->Text = "Settings";//заполение текстом меню settingsToolStripMenuItem на англиском языке

this->rUToolStripMenuItem->Text = "RU";//заполение текстом меню rUToolStripMenuItem на англиском языке

this->bYToolStripMenuItem->Text = "BY";//заполение текстом меню bYToolStripMenuItem на англиском языке

this->eNToolStripMenuItem->Text = "EN";//заполение текстом меню eNToolStripMenuItem на англиском языке

this->greenToolStripMenuItem->Text = "Green";//заполение текстом меню greenToolStripMenuItem на англиском языке

this->blueToolStripMenuItem->Text = "Blue";//заполение текстом меню blueToolStripMenuItem на англиском языке

this->redToolStripMenuItem->Text = "Red";//заполение текстом меню redToolStripMenuItem на англиском языке

this->defaultToolStripMenuItem->Text = "Default";//заполение текстом меню defaultToolStripMenuItem на англиском языке

this->createFileToolStripMenuItem->Text = "Create file";//заполение текстом меню createFileToolStripMenuItem на англиском языке

this->languageToolStripMenuItem->Text = "Language";//заполение текстом меню languageToolStripMenuItem на англиском языке

this->fontToolStripMenuItem->Text = "Font";//заполение текстом меню fontToolStripMenuItem на англиском языке

this->helpToolStripMenuItem->Text = "Help";//заполение текстом меню helpToolStripMenuItem на англиском языке

}

private: System::Void rUToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) // обработчик нажатия на rUToolStripMenuItem, в котором происходит измение языка на русский

{

Language = 3;//присвоение значений в Language

this->label1->Text = "Tекст 1";//заполение текстом надписи label1 на русском языке

this->label2->Text = "Текст 2";//заполение текстом надписи label2 на русском языке

this->label4->Text = "Файлы";//заполение текстом надписи label4 на русском языке

this->button1->Text = "Сравнить";//заполение текстом кнопки button1 на русском языке

this->button2->Text = "Вставить сюда";//заполение текстом кнопки button2 на русском языке

this->button3->Text = "Вставить сюда";//заполение текстом кнопки button3 на русском языке

this->button4->Text = "Поиск";//заполение текстом кнопки button4 на русском языке

this->button5->Text = "Заменить";//заполение текстом кнопки button5 на русском языке

this->fileToolStripMenuItem->Text = L"Файл";//заполение текстом меню fileToolStripMenuItem на англиском языке

this->openFileToolStripMenuItem->Text = L"Открыть";//заполение текстом меню openFileToolStripMenuItem на англиском языке

this->saveToolStripMenuItem->Text = "Сохранить";//заполение текстом меню saveToolStripMenuItem на англиском языке

this->backGroundToolStripMenuItem->Text = "Задний фон";//заполение текстом меню backGroundToolStripMenuItem на англиском языке

this->settingsToolStripMenuItem->Text = "Настройки";//заполение текстом меню settingsToolStripMenuItem на англиском языке

this->rUToolStripMenuItem->Text = L"РУС";//заполение текстом меню rUToolStripMenuItem на англиском языке

this->bYToolStripMenuItem->Text = L"БЕЛ";//заполение текстом меню bYToolStripMenuItem на англиском языке

this->eNToolStripMenuItem->Text = L"АНГЛ";//заполение текстом меню eNToolStripMenuItem на англиском языке

this->greenToolStripMenuItem->Text = L"Зеленый";//заполение текстом меню greenToolStripMenuItem на англиском языке

this->blueToolStripMenuItem->Text = L"Синий";//заполение текстом меню blueToolStripMenuItem на англиском языке

this->redToolStripMenuItem->Text = L"Красный";//заполение текстом меню redToolStripMenuItem на англиском языке

this->defaultToolStripMenuItem->Text = L"Стандартный";//заполение текстом меню defaultToolStripMenuItem на англиском языке

this->createFileToolStripMenuItem->Text = "Создать файл";//заполение текстом меню createFileToolStripMenuItem на англиском языке

this->languageToolStripMenuItem->Text = "Язык";//заполение текстом меню languageToolStripMenuItem на англиском языке

this->fontToolStripMenuItem->Text = "Шрифт";//заполение текстом меню fontToolStripMenuItem на англиском языке

this->helpToolStripMenuItem->Text = "Помощь";//заполение текстом меню helpToolStripMenuItem на англиском языке

}

private: System::Void fileListBox1\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на fileListBox1, в котором происходит выбор файла для открытия

{

String^ URL = folderBrowserDialog1->SelectedPath + "\\" + fileListBox1->SelectedItem->ToString();//декларация и инициализация переменной String адресом файла

URLF = gcnew String(URL);//инициализация глобальной переменной URLF

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Вы выбралi файл " + URLF, "Файл");//показать сообщение с описанием выбраноого файла

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("You selected file " + URLF, "File");//показать сообщение с описание выбраноого файла

}

if (Language == 3)//если Language = 3 => MessageBox на русском языке

{

MessageBox::Show("Вы выбрали файл " + URLF, "Файл");//показать сообщение с описание выбраноого файла

}

}

private: System::Void MainForm\_FormClosing(System::Object^ sender, System::Windows::Forms::FormClosingEventArgs^ e)//обработчик нажатия на кнопку закрытия приложения

{

Application::Exit();//закрытие приложения

}

private: System::Void createFileToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) //обработчик нажатия на createFileToolStripMenuItem, в котором происходит создание txt файла

{

saveFileDialog1->Filter = "Text Files|\*.txt";//фильтация типов файлов

if (saveFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK)//выбор расположения файла на устройстве

{

StreamWriter^ sw = gcnew StreamWriter(saveFileDialog1->FileName);//создание потока записи sw

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Вы ставрылi новы фаЙл у " + saveFileDialog1->FileName, "Створаны файл");//показать сообщение с описанием пути

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("You create new file in " + saveFileDialog1->FileName, "Create file");//показать сообщение с описанием пути

}

if (Language == 3)//если Language = 3 => MessageBox на русском языке

{

MessageBox::Show("Вы создали новый файл в " + saveFileDialog1->FileName, "Создан файл");//показать сообщение с описанием пути

}

sw->Write("");//запись пробела в текстовый документ

sw->Close();//закрытие потока sw

}

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) //обработчик нажатия на button4, в котором происходит поиск слов в тексте

{

try//начало области кода, в которой отслеживаются ошибки

{

if (textBox1->Text == "")//если textBox1 пуст

{

throw 1;//вызов исключения

}

int index = 0;//декларация и инициализация переменной типа int

String^ temp = richTextBox1->Text;//декларация переменной типа String

richTextBox1->Text = "";//очищение richTextBox1

richTextBox1->Text = temp;//декларация переменной типа String

while (index < richTextBox1->Text->LastIndexOf(textBox1->Text))//пока index меньше длины richTextBox2

{

richTextBox1->Find(textBox1->Text, index, richTextBox1->Text->Length, RichTextBoxFinds::None);//использование метода Find

richTextBox1->SelectionBackColor = Color::Red;//изменение цвета заднего фона в найденом слове

index = richTextBox1->Text->IndexOf(textBox1->Text, index) + 1;//декларация переменной типа int

}

int index2 = 0;//декларация и инициализация переменной типа int

String^ temp2 = richTextBox2->Text;//декларация переменной типа String

richTextBox2->Text = "";//очищение richTextBox1

richTextBox2->Text = temp2;//декларация переменной типа String

while (index2 < richTextBox2->Text->LastIndexOf(textBox1->Text))//пока index2 меньше длины richTextBox2

{

richTextBox2->Find(textBox1->Text, index2, richTextBox2->Text->Length, RichTextBoxFinds::None);//использование метода Find

richTextBox2->SelectionBackColor = Color::Red;//изменение цвета заднего фона в найденом слове

index2 = richTextBox2->Text->IndexOf(textBox1->Text, index2) + 1;//декларация переменной типа int

}

}

catch (int)//отлов ошибки типа int

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Радок пуст", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("String is empty", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на русском языке

{

MessageBox::Show("Строка пустая", "Ошибка");//показать сообщение с описание ошибки

}

}

}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на button5, в котором происходит замена слова в тексте

{

try//начало области кода, в которой отслеживаются ошибки

{

if (textBox1->Text == "" || textBox2->Text == "")//провекрка если textBox пуст

{

throw 1;//вызов исключения

}

String^ Find = textBox1->Text;//декларация и инициализация переменной типа String текстом из textBox1

String^ Rep = textBox2->Text;//декларация и инициализация переменной типа String текстом из textBox2

if ((Find->Length) \* 2 == Rep->Length)//проверка на размер слова

{

String^ Part1;//декларация переменной типа String

String^ Part2;//декларация переменной типа String

for (int i = 0; i < Rep->Length / 2; i++)//цикл для записи значений в Part1

{

Part1 += Rep[i];//добавление символа из Rep[i] в Part1

}

for (int i = Rep->Length / 2; i < Rep->Length; i++)//цикл для записи значений в Part2

{

Part2 += Rep[i];//добавление символа из Rep[i] в Part2

}

if (Part1 == Part2)//если Part1 = Part2

{

throw std::logic\_error("");//вызов исключения underflow\_error

}

}

int index = 0;//декларация и инициализация переменной типа int

String^ temp = richTextBox1->Text;//декларация переменной типа String

richTextBox1->Text = "";//очищение richTextBox1

richTextBox1->Text = temp;//декларация переменной типа String

while (index < richTextBox1->Text->LastIndexOf(textBox1->Text))//пока index меньше длины richTextBox2

{

richTextBox1->Find(textBox1->Text, index, richTextBox1->Text->Length, RichTextBoxFinds::None);//использование метода Find

richTextBox1->SelectionBackColor = Color::Red;//изменение цвета заднего фона в найденом слове

index = richTextBox1->Text->IndexOf(textBox1->Text, index) + 1;//декларация переменной типа int

richTextBox1->Focus();//установка фокуса ввода

richTextBox1->SelectedText = textBox2->Text;//запись текста из textBox2 в richTextBox1

}

}

catch (int)//отлов ошибки типа int

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Радок пуст", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("String is empty", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на русском языке

{

MessageBox::Show("Строка пустая", "Ошибка");//показать сообщение с описание ошибки

}

}

catch (std::logic\_error e)//отлов ошибки типа logic\_error

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Выяўлена памылка", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("Error detected", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на русском языке

{

MessageBox::Show("Обнаружена ошибка", "Ошибка");//показать сообщение с описание ошибки

}

}

}

private: System::Void fontToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на fontToolStripMenuItem, в котором происходит выбор шрифта текста

{

FontDialog^ fd = gcnew FontDialog;//декларация окна FondDialog

if (fd->ShowDialog() == System::Windows::Forms::DialogResult::OK)//окртытие окна с выбором шрифта

{

richTextBox1->Font = fd->Font;//использование шрифта на richTextBox1

richTextBox2->Font = fd->Font;//использование шрифта на richTextBox2

}

}

private: System::Void helpToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик создания оконной формы

{

HelpForm^ form = gcnew HelpForm(Language, Color);;//создать новую оконную форму типа класса MainForm конструктором с ;//создать новую оконную форму типа класса MainForm конструктором с вещественным параметромпараметром

form->Show();//показать форму

}

};

}

#include "RegForm.h"

#pragma once

#include <stdexcept>//подключение библиотеки stdexcept для работы с исключениями

#include <string.h>//подключение библиотеки string.h для работы с строками

#include <fstream>//подключение библиотеки fstream дляx чтения файлов

namespace CourseWork {

RegForm(int x, int y)//коструктор принимающий несколько целочисленных параметров

{

InitializeComponent();//метод InitializeComponent()

this->Color = y;//инициализируем глобальную переменную LLanguage

this->Language = x;//инициализируем глобальную переменную Color

}

#pragma endregion

int Color = 0;//декларация и инициализация глобальной переменной типа int для смены языка в программе

int Language = 0;//декларация и инициализация глобальной переменной типа int для смены цвета фона в программе

String^ Message;//декларация глобальной строки

String^ Message2;// декларация глобальной строки

String^ range\_high;//декларация глобальной строки

String^ range\_low;//декларация глобальной строки

String^ range\_same;//декларация глобальной строки

String^ error;//декларация глобальной строки

String^ symbol;//декларация глобальной строки

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)//обработчик нажатия на button1, в котором проходит регистрация пользовтеля в приложении

{

try//начало области кода, в которой отслеживаются ошибки

{

String^ Login = textBox1->Text;//декларация и инициализация строки значениями из TextBox1 которые ввел пользователь

String^ Password = textBox2->Text;//декларация и инициализация строки значениями из TextBox2 которые ввел пользователь

if (Login->Length > 15 | Password->Length > 15)//проверка Login и Password если те больше 15 символов

{

throw std::overflow\_error("");//вызов исключения underflow\_error

}

if (Login->Length < 3 | Password->Length < 3)//проверка Login и Password если те меньше 3 символов

{

throw std::underflow\_error("");//вызов исключения underflow\_error

}

for (int i = 0; i < Login->Length; i++)//проверка Login если в нем содержаться запрещенные символы

{

if (Login[i] == '$' | Login[i] == '&' | Login[i] == '#')//сравнения ячеек строки с символами

{

throw 1;//вызов исключения int

break;//выход из цикла

}

}

for (int i = 0; i < Password->Length; i++)//проверка Password если в нем содержаться запрещенные символы

{

if (Password[i] == '$' | Password[i] == '&' | Password[i] == '#')//сравнения ячеек строки с символами

{

throw 1;//вызов исключения int

break;//выход из цикла

}

}

if (Login == Password)//проверка если Login == Password

{

throw 0.1;//вызов исключения

}

int lattice = 0;//декларация и инициализация переменной типа int

bool pass;//декларация переменной типа bool

StreamReader^ check = gcnew System::IO::StreamReader("Data.txt");//создние потока для чтения check из файла Data.txt

{

while(!check ->EndOfStream)//если поток не дошел до конца, то выполнят условие

{

String^ Text = check->ReadLine();//декларация и инициализация строковой переменной с заполнение строкой которую читает программа

if (Text == nullptr)//если строка Text пустая

{

check->Close();//закрытие потока check

break;//выход из цикла

}

for (int i = 0; i < Text->Length; i++)//создание цикла для чтения посимвольная строки Text

{

if (Text[i] == '#')//сравнение ячейки Text с индексом i если они равна #

{

String^ Log;//декларация строки строковой переменно Log

lattice = i;//располжение # в строке

for (int j = 0; j < lattice; j++)//в цикле заполняем строку Log значениями из Text

{

Log += Text[j];//Заполнение строки

}

if (Log == Login)//сравнение Log(Логина из строки в файле) с Login(Логин который ввел пользователь

{

pass = true;//присвоение переменной значения true

break;//выход из цикла

}

}

}

}

}

check->Close();//закрытие потока check

if (pass == true)//если text == true то логин занят

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("Дадзены лагiн ужо заняты.", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 1 => MessageBox на англиском языке

{

MessageBox::Show("This login is already occupied.", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 1 => MessageBox на русском языке

{

MessageBox::Show("Данный логин уже занят.", "Ошибка");//показать сообщение с описание ошибки

}

}

else//иначе

{

StreamWriter^ write = File::AppendText("Data.txt");//создание txt файла, если его не нашла программа

write->WriteLine(Login + "#" + Password);//занесение логина и пароля с разделяющим знаком #

MessageBox::Show(Message, Message2);//сообщение о том что файл занесен

write->Close();//закрытие потока write

RegForm::Close();//закрытие формы RegForm

}

}

catch (System::IO::FileNotFoundException^ e)//отлов ошибки типа FileNotFoundException

{

if (Language == 1)//если Language = 1 => MessageBox на белорусском языке

{

MessageBox::Show("База пароляў пустая, зарэгіструйцеся!", "Памылка");//показать сообщение с описание ошибки

}

if (Language == 2)//если Language = 2 => MessageBox на англиском языке

{

MessageBox::Show("Password database is empty, register!", "Error");//показать сообщение с описание ошибки

}

if (Language == 3)//если Language = 3 => MessageBox на русском языке

{

MessageBox::Show("База паролей пуста, зарегистрируйтесь!", "Ошибка");//показать сообщение с описание ошибки

}

std::ofstream error("Data.txt");//создание txt файла с именем Data.txt

error.close();//закрытие потока error

}

catch (int)//отлов ошибки типа int

{

MessageBox::Show(symbol, Message2);//показать сообщение с описание ошибки

}

catch (std::overflow\_error e)//отлов ошибки типа overflow\_error

{

MessageBox::Show(range\_high, Message2);//показать сообщение с описание ошибки

}

catch (std::underflow\_error e)//отлов ошибки типа underflow\_error

{

MessageBox::Show(range\_low, Message2);//показать сообщение с описание ошибки

}

catch (double) //отлов ошибки типа double

{

MessageBox::Show(range\_same, error);//показать сообщение с описание ошибки

}

}

private: System::Void RegForm\_Load(System::Object^ sender, System::EventArgs^ e)//метот-обработчик события загрузки оконной формы

{

if (Language == 1)//если Language = 1 => установка белорусского языка

{

symbol = "Уведзены забароненыя сімвалы";//заполение текстом строки на белорусском языке

error = "Памылка";//заполение текстом строки на белорусском языке

range\_same = "Лагiн i пароль падобныя. Паутарыце спробу.";//заполение текстом строки на белорусском языке

Message = "Рэгистрация пройдзена";//заполение текстом строки на белорусском языке

Message2 = "Рэгистрация";//заполение текстом строки на белорусском языке

range\_high = "Лагiн або пароль больш за 15 сiмвалау";//заполение текстом строки на белорусском языке

range\_low = "Лагiн або пароль менш за 15 сiмвалау";//заполение текстом строки на белорусском языке

this->button1->Text = "Зарэгистравацца";//заполение текстом кнопки button1 на белорусском языке

this->label1->Text = "Лагiн";//заполение текстом надписи label1 на белорусском языке

this->label2->Text = "Пароль";//заполение текстом надписи label2 на белорусском языке

}

if (Language == 2)//если Language = 2 => установка англиского языка

{

symbol = "Forbidden characters entered";//заполениетекстом строки на англиском языке

error = "Error";//заполениетекстом строки на англиском языке

range\_same = "Login and password are similar. Try again.";//заполение текстом строки на англиском языке

Message = "Registration Complited";//заполение текстом строки на англиском языке

Message2 = "Registration";//заполение текстом строки на англиском языке

range\_high = "Login or password more that 15 character.";//заполение текстом строки на англиском языке

range\_low = "Login or password less that 3 character.";//заполение текстом строки на англиском языке

this->button1->Text = "Register";//заполение текстом кнопки button1 на англиском языке

this->label1->Text = "Login";//заполение текстом надписи label1 на англиском языке

this->label2->Text = "Password";//заполение текстом надписи label2 на англиском языке

}

if (Language == 3)//если Language = 3 => установка русского языка

{

symbol = "Введены запрещенные символы";//заполение текстом строки на русском языке

error = "Ошибка";//заполение текстом строки на русском языке

range\_same = "Логин и пароль схожи. Повторите попытку.";//заполение текстом строки на русском языке

Message = "Регистрация пройдена";//заполение текстом строки на русском языке

Message2 = "Регистрация";//заполение текстом строки на русском языке

range\_high = "Логин или пароль больше 15 символов";//заполение текстом строки на русском языке

range\_low = "Логин или пароль меньше 3 символов";//заполение текстом строки на русском языке

this->button1->Text = "Зарегистрироваться";//заполение текстом кнопки button1 на русском языке

this->label1->Text = "Логин";//заполение текстом надписи label1 на русском языке

this->label2->Text = "Пароль";//заполение текстом надписи label2 на русском языке

}

if (Color == 0)//если Color = 0 => установка фонового цвета "Кнопочный серый"

{

this->BackColor = SystemColors::ButtonFace;//назначить фоновым цветом ЭТОЙ оконной формы ситемный цвет "кнопочный серый"

this->label1->ForeColor = Color::Black;//назначить цвет надписи Label1 "Черный цвет"

this->label2->ForeColor = Color::Black;//назначить цвет надписи Label2 "Черный цвет"

}

if (Color == 1) //если Color = 1 => установка фонового цвета "Синий"

{

this->BackColor = Color::Blue;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Синий"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

}

if (Color == 2) //если Color = 2 => установка фонового цвета "Зеленый"

{

this->BackColor = Color::Green;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Зеленый"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

}

if (Color == 3) //если Color = 3 => установка фонового цвета "Красный"

{

this->BackColor = Color::Red;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Красный"

this->label1->ForeColor = Color::White;//назначить цвет надписи Label1 "Белый"

this->label2->ForeColor = Color::White;//назначить цвет надписи Label2 "Белый"

}

}

};

}

#include "HelpForm.h"

#pragma once

namespace CourseWork {

HelpForm(int x, int y)//коструктор принимающий несколько целочисленных параметров

{

InitializeComponent();//коструктор принимающий несколько целочисленных параметров

this->Language = x;//инициализируем глобальную переменную Language

this->Color = y;//инициализируем глобальную переменную Color

}

int Color = 0;//декланируется и инициализируется глобальная переменная типа int для смены цвета в программе

int Language = 0;//декланируется и инициализируется глобальная переменная типа int для смены языка в программе

private: System::Void HelpForm\_Load(System::Object^ sender, System::EventArgs^ e)//метот-обработчик события загрузки оконной формы

{

if (Color == 0)//если Color = 0 => установка фонового цвета "Кнопочный серый"

{

this->BackColor = SystemColors::ButtonFace;//назначить фоновым цветом ЭТОЙ оконной формы ситемный цвет "кнопочный серый"

}

if (Color == 1)//если Color = 1 => установка фонового цвета "Синий"

{

this->BackColor = Color::Blue;;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Синий"

}

if (Color == 2)//если Color = 2 => установка фонового цвета "Зеленый"

{

this->BackColor = Color::Green;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Зеленый"

}

if (Color == 3)//если Color = 3 => установка фонового цвета "Красный"

{

this->BackColor = Color::Red;//назначить фоновым цветом ЭТОЙ оконной формы цвет "Красный"

}

if (Language == 1)//если Language = 1 => установка белорусского языка

{

StreamReader^ read = gcnew System::IO::StreamReader("HelpBy.txt");//создвние потока sr для чтение HelpBy.txt

richTextBox1->Text = read->ReadToEnd();//добавить в richTextBox1 весь текст из HelpBy.txt

read->Close();//закртыть поток read

}

if (Language == 2)//если Language = 2 => установка англиского языка

{

StreamReader^ read = gcnew System::IO::StreamReader("HelpEn.txt");//создвние потока sr для чтение HelpEn.txt

richTextBox1->Text = read->ReadToEnd();//добавить в richTextBox1 весь текст из HelpBy.txt

read->Close();//закртыть поток read

}

if (Language == 3)//если Language = 3 => установка русского языка

{

StreamReader^ read = gcnew System::IO::StreamReader("HelpRu.txt");//создвние потока sr для чтение HelpRu.txt

richTextBox1->Text = read->ReadToEnd();//добавить в richTextBox1 весь текст из HelpBy.txt

read->Close();//закртыть поток read

}

Приложение Б

(справочное)

Результат работы программы

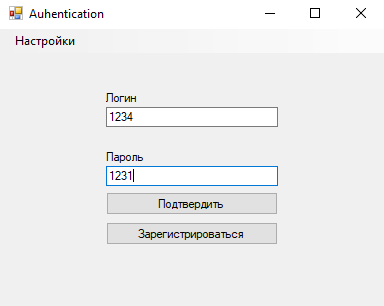


Рисунок Б.1 – Результат действия «Запустить приложение AntiplagiarismFix.exe»

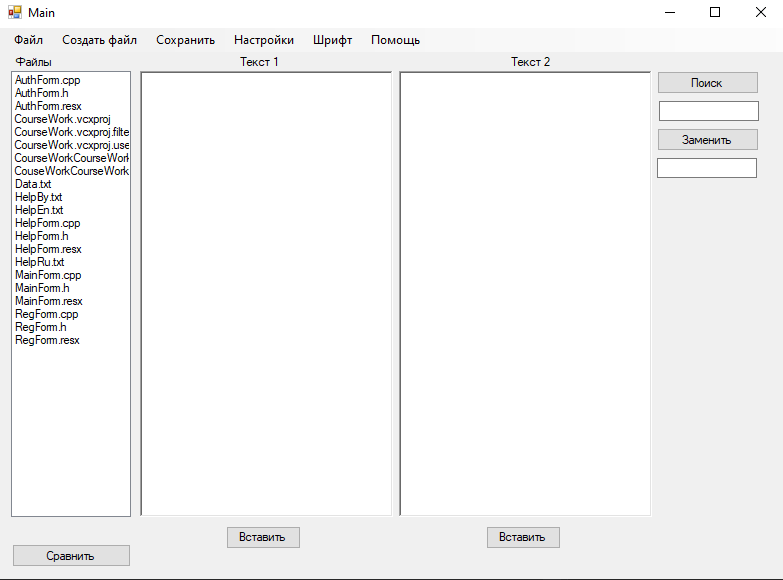


Рисунок Б.2 – Результат действия «Авторизоваться в приложении»

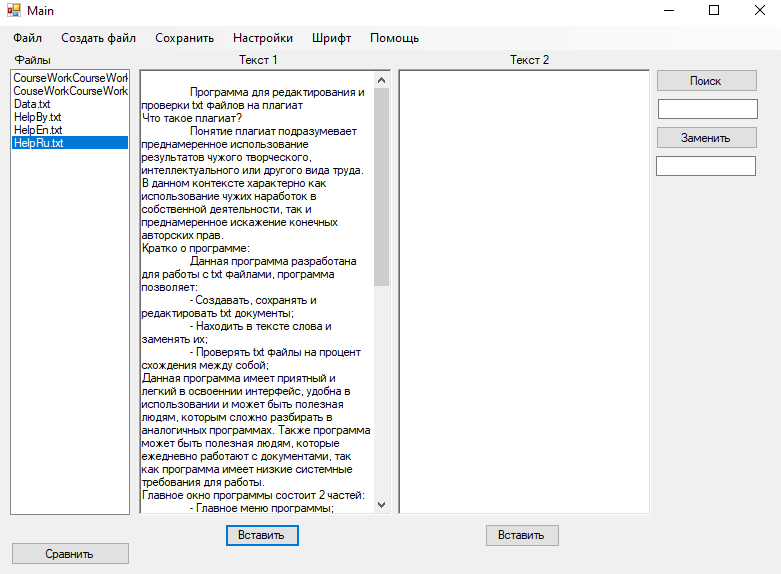


Рисунок Б.3 – Результат действия «Открыть файл»

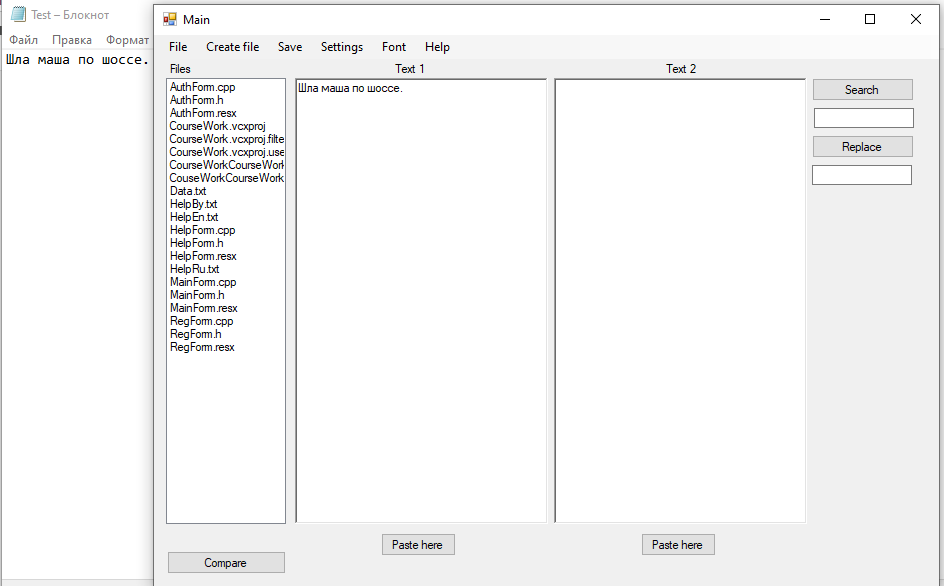


Рисунок Б.4 – Результат функции «Сохранение файла»



КБиП

У

Т. Контр.

Разраб.

Самойлов А.С.

Провер.

Шаляпин Ю.В.

Реценз.

Утверд.

Н. Контр.

*Программа для обработки текстовых файлов с функцией антиплагиата*

№ докум.

Подп.

Дата

Изм.

Лист

Масса

Лист

Масштаб

Лист 1

Листов 1

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

*КП Т.0910015.401 ГЧ*

*КП Т.0910015.401 ГЧ*

Приложение В.

*Блок-схема работы программы (общая)*



*КП Т.0910015.401 ГЧ*

КБиП

У

Т. Контр.

Разраб.

Самойлов А.С.

Провер.

Шаляпин Ю.В.

Реценз.

Утверд.

Н. Контр.

*Программа для обработки текстовых файлов с функцией антиплагиата*

Блок-схема работы алгоритма антиплагиата

№ докум.

Подп.

Дата

Изм.

Лист

Масса

Лист

Масштаб

Лист 1

Листов 2

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

*КП Т.0910015.401 ГЧ*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| *Изм.* | *Лист* | *№ докум.* | *Подп.* | *Дата* | *КП Т.0910015.401 ГЧ* | *Лист* |
|  |  |  |  |  | 2  *2* |
|  |  |  |  |  |

**Удостоверяющий лист**

электронного документа – курсовой проект

Тема КППрограмма для обработки текстовых файлов с функцией антиплагиата

Обозначение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ КП Т.091015.401\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Разработчик Самойлов А.С. Руководитель Шаляпин Ю.В.

(Ф.И.О.) (Ф.И.О.)

Подписи лиц, ответственных за разработку электронного документа

|  |  |  |
| --- | --- | --- |
| Состав электронного документа | Разработчик | Руководитель |
| Пояснительная записка (на бумажном носителе формата А4), ПЗ\_СамойловТ091.docх |  |  |
| ГЧ, Блок-схема работы программы(общая).docx |  |  |
| ГЧ, Блок-схема работы алгоритма составления сметы.docx |  |  |
| Папка с проектом «CourseWork» |  |  |
| Установочный пакет программного средства «Setup.exe» |  |  |
| Тип носителя: оптический компакт-диск |  |  |

**Этикетка**

для курсового проекта

**Курсовой проект**

Тема « Программа для обработки текстовых файлов с функцией антиплагиата»

КП Т.091015.401

Разработан

Утвержден

Разработчик Самойлов А.С.

Руководитель: Шаляпин Ю.В.

Технические средства: Процессор 1.3 ГГц, ОЗУ 6 ГБ, 500 ГБ свободного места

Программные средства: MS Visual Studio 2019 Community, MS Word, MS Excel,

**Состав документа:**

Пояснительная записка – ПЗ\_СамойловТ091.docх

Графическая часть – Блок-схема работы программы.docx, Блок-схема работы алгоритма составления сметы.docx.

Папка с проектом – CourseWork

Установочный пакет программного средства – Setup.exe