Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное   
образовательное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ЛАБОРАТОРНАЯ РАБОТА №3. СТРОКИ   
И РАБОТА С ФАЙЛАМИ

Отчет по лабораторной работе №3 по учебной дисциплине «Алгоритмические языки и программирование»

по специальности 09.03.01 Информатика и вычислительная техника

|  |  |
| --- | --- |
|  | Руководитель  / И. Ю. Кулаков  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |
|  | Студент группы 3091  / Р. А. Михайлов  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

1. **Постановка задачи**

Необходимо реализовать программу для работы с текстовыми файлами, обработки их содержимого и сделать отчет.

* 1. **Наименование задачи** – Лабораторная работа №3. Строки и работа с файлами.
  2. **Словесное описание**:

Напишите программу для работы с текстовыми файлами с возможностью:

1. Чтения определенного файла;
2. Редактирования содержимого файла (по вариантам);
3. Сохранение отредактированной информации в файл;
4. Создание нового файла;
5. Выбора файлов для использования.

Разработайте графический интерфейс для приложения, включающий в себя:

1. Меню выбора файла для открытия;
2. Поле для отображения текста выбранного файла;
3. Кнопка сохранения файла
4. Возможность редактирования файла:

* После каждого знака вопроса поставить без пробелов символ «%»;
* Подсчитать количество букв в тексте и сравнить его с количеством цифр в тексте, написать в начало текста «Буквы», если букв больше, «Цифры» в ином случае.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объект программы | Имя этого объекта в программе | Тип данных | Диапазоны представления | Простая/  структура | Вход/  выход/  Константа |
| Окно текста | This->text->Text | String^ | - | Структура | Вход/ Выход |

Внешние спецификации данных

* 1. **Внешние спецификации функций программы**

В программе должен быть реализован выбор файла, его открытие, сохранение информации в данный файл, сохранение информации в выбранный файл, конвертация из формата файла txt в bin и обратно, переименование текущего файла.

Должны быть реализованы функции редактирования: вставка знака процента после вопросительного знака; определение чего больше в файле цифр или букв с последующей вставкой соответствующего в начало файла слова «Буквы» или «Цифры».

* 1. **Математическая формулировка задачи**

Ограничение количества вводимых в поле ввода символов – 1073741791 (максимальная длина строки в данном языке программирования)

При переименовании файла максимальная длина нового имени не должна превышать 250 символов, так же при переименовании файла нельзя использовать некоторые специальные символы, которые указаны в информационном окне приложения. Ограничения установлены в соответствии с требованиями операционной системой.

* 1. **Спецификации интерфейса**

Для написания графического интерфейса была использована библиотека Windows Forms.

Плюсы интерфейса заключаются в том, что реализован контролируемый ввод, при попытке ввода некорректных данных срабатывает функция контроля ввода.

Реализована обработка повторного нажатия ключевых кнопок, что предотвращает некорректный вывод данных.

В приложении предусмотрена встроенное тестирование при запуске программы, и отдельная кнопка для тестирования работоспособности ключевых расчетных функций.

В приложении реализована работа со строками и файлами, ввод/вывод, очистка полей, кнопки, выводящие справочную информацию о работы программы.

* 1. **Внешние данные тестирования**

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | T01 |
| Заголовок | | | Проверка выхода из приложения |
| Предусловие | | | Приложение не запущено |
| Ожидаемый результат | | | Успешный выход из приложения |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Запуск LAB\_3.exe | Открытие главного окна приложения | Pass |
| 2 | Нажатие на кнопку «Выход» | Выход из приложениы | Pass |
| Результат | | | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | T02 |
| Заголовок | | | Проверка вывода имени файла и его расширения |
| Предусловие | | | Приложение запущено |
| Ожидаемый результат | | | Успешный вывод информации о файле |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Нажатие «Файл» | Открытие меню файла | Pass |
| 2 | Нажатие на кнопку «Открыть» | Вывод окна выбора файла | Pass |
| 3 | Выбрать файл | Закрытие окна выбора файла, вывод имени и расширения файла | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | T03 |
| Заголовок | | | Проверка ввода нового имени файла |
| Предусловие | | | Приложение запущено, файл открыт |
| Ожидаемый результат | | | Вывод окна ошибки |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Нажать «Файл» | Открытие меню файла | Pass |
| 2 | Нажатие на кнопку «Переименовать» | Вывод окна переименования файла | Pass |
| 3 | Ввод некорректного имени файла | Вывод окна ошибки | Pass |
| Результат | | | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | T04 |
| Заголовок | | | Проверка работоспособности функции count() |
| Предусловие | | | Приложение запущено, открыт непустой файл из цифр |
| Ожидаемый результат | | | Добавление в начало текста подстроки «Цифры» |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Нажать «Редактирование» | Открытие меню редактирования | Pass |
| 2 | Нажать на кнопку «Больше букв или цифр» | Добавление в начало текста подстроки «Цифры» | Pass |
| Результат | | | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | T05 |
| Заголовок | | | Проверка работоспособности функции count() |
| Предусловие | | | Приложение запущено, открыт непустой файл, состоящий из вопросительных знаков |
| Ожидаемый результат | | | Вставка после каждого символа «?» символа «%» |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Нажать «Редактирование» | Открытие меню редактирования | Pass |
| 2 | Нажать на кнопку «%» после «?» | Вставка после каждого вопросительного знака символа процента | Pass |
| Результат | | | Pass |

* 1. **Пример работающей программы**

Изображение выглядит как снимок экрана, компьютер, текст, программное обеспечение

Автоматически созданное описание

Рисунок 1. Окно приложения

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2. Информационное окно

1. **Проектирование задачи**

Реализовать приложение с графическим интерфейсом, использую библиотеку Windows Forms, для работы со строками и файлами. Модульное тестирование, используя фреймворк CppUnitTest и встроенное тестирование, при помощи процедур и функций, проводимое как при запуске программы, так и при нажатии отдельной кнопки.

* 1. **Наименование программы** –LAB\_3.exe
  2. **Уточненное словесное описание задачи**

Напишите программу для работы с текстовыми файлами с возможностью:

1. Чтения определенного файла;
2. Редактирования содержимого файла (по вариантам);
3. Сохранение отредактированной информации в файл;
4. Создание нового файла;
5. Выбора файлов для использования.

Разработайте графический интерфейс для приложения, включающий в себя:

1. Меню выбора файла для открытия;
2. Поле для отображения текста выбранного файла;
3. Кнопка сохранения файла
4. Возможность редактирования файла:

* После каждого знака вопроса поставить без пробелов символ «%»;
* Подсчитать количество букв в тексте и сравнить его с количеством цифр в тексте, написать в начало текста «Буквы», если букв больше, «Цифры» в ином случае.
  1. **Выбор метода решения поставленной задачи**

Описываем методику вашего решения, обычно описываются существующие методы.

При открытии файла появляется всплывающее окно с выбором фала для открытия, после выбора содержимое файла выводится на экран, в поле текста файла.

При сохранении информации в файл, текс, находящийся в поле теста, записывается в открытый файл, или выбранный файл, в случае выбора пункта «Сохранить в».

При закрытии файла поле текста файла, поле имени файла, и поле расширения файла очищаются, для них устанавливается значение по умолчанию.

При переименовании файла, после ввода нового имени файла, файл со старым именем удаляется, а на его место записывается новый файл, с той же информаций, но другим именем.

При создании файла создается новый файл, с указанным именем, посредством открытия файла по указанному пути, без вывода содержимого на экран (в поле текста).

При редактировании «процент после вопроса» во входной строке считается количество буквенных символов и цифр, после чего происходит сравнение полученных значений.

При редактировании «Цифры или Буквы» программа проходится по строке через цикл, вставляя после каждого вопросительного знака символа процента. Так же есть обработка граничных случаев, строка состоит из одного вопросительного знака, и вопросительный знак стоит в конце.

При конвертации файла из txt в bin и обратно, файл с текущим расширение удаляется, а на его место записывается новый файл, с той же информацией, но с другим расширением. Перезапись информации происходит при помощи соответствующих объектов (BinaryWriter для bin и StreamWriter для txt).

* 1. **Уточненные глобальные данные программы и пользовательские типы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объект программы | Имя этого объекта в программе | Характеристики | | | Как используется в программе |
| Тип данных | Диапазоны представления | Прост/структура/ константа |
| LAB\_3.h | open\_file\_name | String^ | ­­­- | Структура | Хранит полный путь открытого файла |

* 1. **Декомпозиция функций**

Описывается каждая функция и процедура в программе

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Назначение | Имя | Параметры | | Процедура/ функция | Тестируется/  не тестируется |
| Вход:  тип | Выход: тип |
| Определение, чего больше в тексте, букв или цифр | count | - | void | Процедура | Нет |
| Вставка символа процента после каждого вхождения впоросительного знака | insert\_percent | String^ str | void | Процедура | Нет |
| Вывод имя файла и его расширения | file\_info | String^ info | void | Процедура | Нет |
| Чтение из файла | read\_from\_file | String^ filename | void | Процедура | Нет |
| Запись в файл | write\_to\_file | String^ filename | Void | Процедура | Нет |
| Создание файла | create\_file | String^ filename | Void | Процедура | Нет |
| Удаление файла | delete\_file | String^ filename | void | Процедура | Нет |
| Переименовать файл | rename | String^ new\_name | Void | Процедура | Нет |
| Действия при выборе создания файла | создатьToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Действия при выборе открытия файла | открытьToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Действия при выборе сохранения файла | сохранитьToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Действия при выборе редактирования (% после ?) | toolStripMenuItem2\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Действия при выборе редактирования (больше букв или цифр) | большеБуквИлиЦифрToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Действия при выборе конвертации .txt в .bin | txtВbinToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Действия при выборе очистки полей | очиститьПолеToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Действия при выборе переименования | переименоватьToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Проверка количества символов в названии | max\_quantity | String^ str | bool | Процедура | Нет |
| Проверка корректности введенных символов | valid\_input | String^ str | bool | Процедура | Нет |
| Дейстивя при выборе сохранения в | сохранитьВФайлToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Действия при выборе О программе | оПрограммеToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Функция тестирования процедуры count() | count\_test | - | int | Функция | Нет |
| Функция тестирования процедуры insetr\_percent() | insert\_test | - | int | Функция | Нет |
| Функция тестирования процедуры file\_info() | file\_info\_test | - | int | Функция | Нет |
| Действия при выборе Тест | тестToolStripMenuItem\_Click | System::Object^ sender, System::EventArgs^ e | System::Void | Процедура | Нет |
| Процедура тестирования программы при вызове конструктора | constructor\_test | - | void | Процедура | Нет |

* 1. **Блок-схема**

Блок-схема функции count() в Приложении A.  
Блок-схема функции insert\_percent() в Приложении Б.

Блок-схема процедуры valid\_input() в Приложении В.

Блок-схема процедуры rename() в Приложении Г.

Блок-схема процедуры сохранитьToolStripMenuItem\_Click() в Приложении Д.

* 1. **Алгоритмизация**

Листинг файла LAB\_3.h в Приложение Е.

Листинг файла LAB\_3.cpp в Приложение Ж.

Листинг файла UnitTestLAB\_3.cpp в Приложение И.

Листинг файла Functions.h в Приложение К.

Приложении A

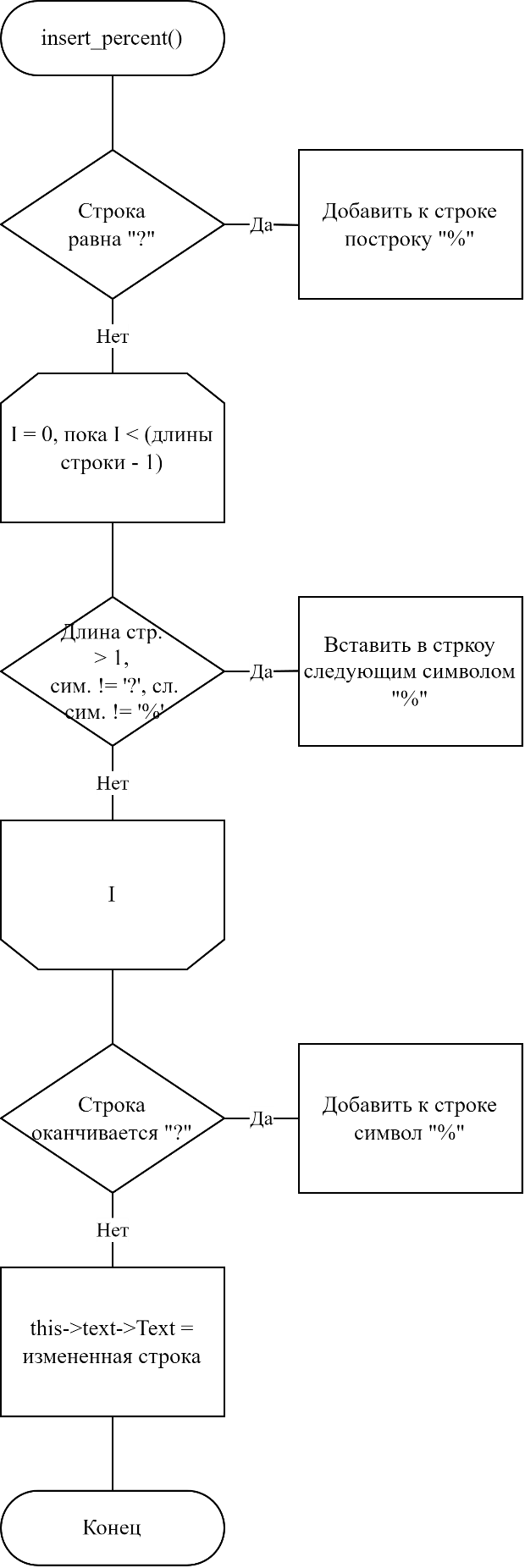
Блок-схема функции count()

Изображение выглядит как текст, снимок экрана, черно-белый, Шрифт

Автоматически созданное описание

Приложении Б

Блок-схема функции insert\_percent()



Приложении В

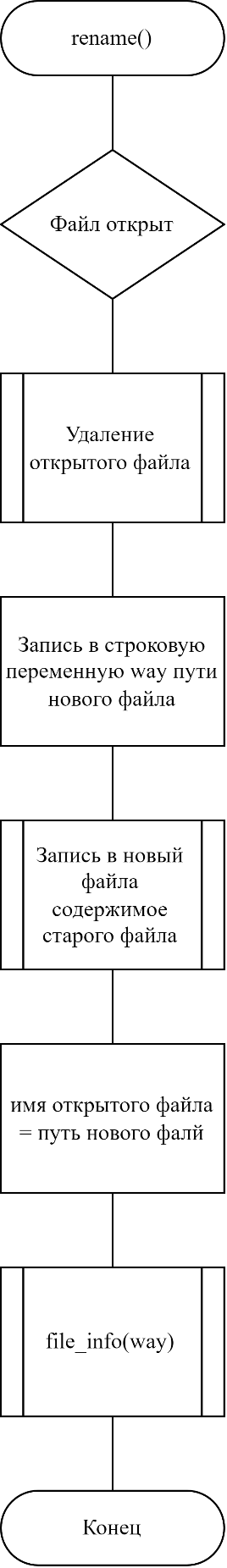
Блок-схема функции valid\_input()

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Приложении Г

Блок-схема процедуры rename()



Приложении Д

Блок-схема процедуры сохранитьToolStripMenuItem\_Click()

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Приложение Е

Листинг файла LAB\_3.h

#pragma once

namespace LAB3 {

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;

using namespace System::Text;

using namespace Microsoft::VisualBasic;

/// <summary>

/// Сводка для LAB\_3

/// </summary>

public ref class LAB\_3 : public System::Windows::Forms::Form

{

public:

LAB\_3(void)

{

InitializeComponent();

SaveFileDialog->Filter = "Текстовый файл (\*.txt)|\*.txt|Бинарный файл (\*.bin)|\*.bin";

OpenFileDialog->Filter = "Текстовый файл (\*.txt)|\*.txt|Бинарный файл (\*.bin)|\*.bin";

constructor\_test();

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~LAB\_3()

{

if (components)

{

delete components;

}

}

// Пользовательские переменны

private: String^ open\_file\_name = "";

protected:

private: System::Windows::Forms::OpenFileDialog^ OpenFileDialog;

private: System::Windows::Forms::SaveFileDialog^ SaveFileDialog;

private: System::Windows::Forms::TextBox^ text;

private: System::Windows::Forms::MenuStrip^ menuStrip1;

private: System::Windows::Forms::ToolStripMenuItem^ файлToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ редактированиеСодержимогоToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ txtВbinToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ создатьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ открытьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ сохранитьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ toolStripMenuItem2;

private: System::Windows::Forms::ToolStripMenuItem^ большеБуквИлиЦифрToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ очиститьПолеToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ переименоватьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ сохранитьВФайлToolStripMenuItem;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ name\_of\_file;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ extension;

private: System::Windows::Forms::ToolStripMenuItem^ оПрограммеToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ тестToolStripMenuItem;

private: System::ComponentModel::IContainer^ components;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(LAB\_3::typeid));

this->OpenFileDialog = (gcnew System::Windows::Forms::OpenFileDialog());

this->SaveFileDialog = (gcnew System::Windows::Forms::SaveFileDialog());

this->text = (gcnew System::Windows::Forms::TextBox());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->файлToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->создатьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->открытьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->сохранитьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->сохранитьВФайлToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->переименоватьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->очиститьПолеToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->редактированиеСодержимогоToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->toolStripMenuItem2 = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->большеБуквИлиЦифрToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->txtВbinToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->оПрограммеToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->label1 = (gcnew System::Windows::Forms::Label());

this->name\_of\_file = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->extension = (gcnew System::Windows::Forms::Label());

this->тестToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->menuStrip1->SuspendLayout();

this->SuspendLayout();

//

// text

//

this->text->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((((System::Windows::Forms::AnchorStyles::Top | System::Windows::Forms::AnchorStyles::Bottom)

| System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->text->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->text->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->text->Location = System::Drawing::Point(12, 31);

this->text->Multiline = true;

this->text->Name = L"text";

this->text->ScrollBars = System::Windows::Forms::ScrollBars::Vertical;

this->text->Size = System::Drawing::Size(717, 558);

this->text->TabIndex = 0;

//

// menuStrip1

//

this->menuStrip1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->menuStrip1->GripMargin = System::Windows::Forms::Padding(2, 2, 0, 2);

this->menuStrip1->ImageScalingSize = System::Drawing::Size(24, 24);

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(5) {

this->файлToolStripMenuItem,

this->редактированиеСодержимогоToolStripMenuItem, this->txtВbinToolStripMenuItem, this->оПрограммеToolStripMenuItem, this->тестToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(741, 32);

this->menuStrip1->TabIndex = 8;

this->menuStrip1->Text = L"menuStrip1";

//

// файлToolStripMenuItem

//

this->файлToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(6) {

this->создатьToolStripMenuItem,

this->открытьToolStripMenuItem, this->сохранитьToolStripMenuItem, this->сохранитьВФайлToolStripMenuItem, this->переименоватьToolStripMenuItem,

this->очиститьПолеToolStripMenuItem

});

this->файлToolStripMenuItem->Name = L"файлToolStripMenuItem";

this->файлToolStripMenuItem->Size = System::Drawing::Size(68, 28);

this->файлToolStripMenuItem->Text = L"Файл";

//

// создатьToolStripMenuItem

//

this->создатьToolStripMenuItem->Name = L"создатьToolStripMenuItem";

this->создатьToolStripMenuItem->Size = System::Drawing::Size(230, 34);

this->создатьToolStripMenuItem->Text = L"Создать";

this->создатьToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::создатьToolStripMenuItem\_Click);

//

// открытьToolStripMenuItem

//

this->открытьToolStripMenuItem->Name = L"открытьToolStripMenuItem";

this->открытьToolStripMenuItem->Size = System::Drawing::Size(230, 34);

this->открытьToolStripMenuItem->Text = L"Открыть";

this->открытьToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::открытьToolStripMenuItem\_Click);

//

// сохранитьToolStripMenuItem

//

this->сохранитьToolStripMenuItem->Name = L"сохранитьToolStripMenuItem";

this->сохранитьToolStripMenuItem->Size = System::Drawing::Size(230, 34);

this->сохранитьToolStripMenuItem->Text = L"Сохранить";

this->сохранитьToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::сохранитьToolStripMenuItem\_Click);

//

// сохранитьВФайлToolStripMenuItem

//

this->сохранитьВФайлToolStripMenuItem->Name = L"сохранитьВФайлToolStripMenuItem";

this->сохранитьВФайлToolStripMenuItem->Size = System::Drawing::Size(230, 34);

this->сохранитьВФайлToolStripMenuItem->Text = L"Сохранить в";

this->сохранитьВФайлToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::сохранитьВФайлToolStripMenuItem\_Click);

//

// переименоватьToolStripMenuItem

//

this->переименоватьToolStripMenuItem->Name = L"переименоватьToolStripMenuItem";

this->переименоватьToolStripMenuItem->Size = System::Drawing::Size(230, 34);

this->переименоватьToolStripMenuItem->Text = L"Переименовать";

this->переименоватьToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::переименоватьToolStripMenuItem\_Click);

//

// очиститьПолеToolStripMenuItem

//

this->очиститьПолеToolStripMenuItem->Name = L"очиститьПолеToolStripMenuItem";

this->очиститьПолеToolStripMenuItem->Size = System::Drawing::Size(230, 34);

this->очиститьПолеToolStripMenuItem->Text = L"Закрыть файл";

this->очиститьПолеToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::очиститьПолеToolStripMenuItem\_Click);

//

// редактированиеСодержимогоToolStripMenuItem

//

this->редактированиеСодержимогоToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(2) {

this->toolStripMenuItem2,

this->большеБуквИлиЦифрToolStripMenuItem

});

this->редактированиеСодержимогоToolStripMenuItem->Name = L"редактированиеСодержимогоToolStripMenuItem";

this->редактированиеСодержимогоToolStripMenuItem->Size = System::Drawing::Size(153, 28);

this->редактированиеСодержимогоToolStripMenuItem->Text = L"Редактирование";

//

// toolStripMenuItem2

//

this->toolStripMenuItem2->Name = L"toolStripMenuItem2";

this->toolStripMenuItem2->Size = System::Drawing::Size(284, 34);

this->toolStripMenuItem2->Text = L"\"%\" после \"\?\"";

this->toolStripMenuItem2->Click += gcnew System::EventHandler(this, &LAB\_3::toolStripMenuItem2\_Click);

//

// большеБуквИлиЦифрToolStripMenuItem

//

this->большеБуквИлиЦифрToolStripMenuItem->Name = L"большеБуквИлиЦифрToolStripMenuItem";

this->большеБуквИлиЦифрToolStripMenuItem->Size = System::Drawing::Size(284, 34);

this->большеБуквИлиЦифрToolStripMenuItem->Text = L"Больше букв или цифр";

this->большеБуквИлиЦифрToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::большеБуквИлиЦифрToolStripMenuItem\_Click);

//

// txtВbinToolStripMenuItem

//

this->txtВbinToolStripMenuItem->Name = L"txtВbinToolStripMenuItem";

this->txtВbinToolStripMenuItem->Size = System::Drawing::Size(98, 28);

this->txtВbinToolStripMenuItem->Text = L"txt <=> bin";

this->txtВbinToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::txtВbinToolStripMenuItem\_Click);

//

// оПрограммеToolStripMenuItem

//

this->оПрограммеToolStripMenuItem->Name = L"оПрограммеToolStripMenuItem";

this->оПрограммеToolStripMenuItem->Size = System::Drawing::Size(124, 28);

this->оПрограммеToolStripMenuItem->Text = L"О программе";

this->оПрограммеToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::оПрограммеToolStripMenuItem\_Click);

//

// label1

//

this->label1->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left));

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->Location = System::Drawing::Point(12, 592);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(100, 20);

this->label1->TabIndex = 9;

this->label1->Text = L"Имя файла:";

//

// name\_of\_file

//

this->name\_of\_file->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left));

this->name\_of\_file->AutoSize = true;

this->name\_of\_file->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->name\_of\_file->Location = System::Drawing::Point(77, 593);

this->name\_of\_file->Name = L"name\_of\_file";

this->name\_of\_file->Size = System::Drawing::Size(0, 20);

this->name\_of\_file->TabIndex = 10;

//

// label2

//

this->label2->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Right));

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label2->Location = System::Drawing::Point(544, 593);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(107, 20);

this->label2->TabIndex = 11;

this->label2->Text = L"Расширение:";

//

// extension

//

this->extension->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Right));

this->extension->AutoSize = true;

this->extension->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->extension->Location = System::Drawing::Point(613, 593);

this->extension->Name = L"extension";

this->extension->Size = System::Drawing::Size(0, 20);

this->extension->TabIndex = 12;

//

// тестToolStripMenuItem

//

this->тестToolStripMenuItem->Name = L"тестToolStripMenuItem";

this->тестToolStripMenuItem->Size = System::Drawing::Size(60, 28);

this->тестToolStripMenuItem->Text = L"Тест";

this->тестToolStripMenuItem->Click += gcnew System::EventHandler(this, &LAB\_3::тестToolStripMenuItem\_Click);

//

// LAB\_3

//

this->AutoScaleDimensions = System::Drawing::SizeF(9, 20);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(741, 619);

this->Controls->Add(this->extension);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->name\_of\_file);

this->Controls->Add(this->text);

this->Controls->Add(this->menuStrip1);

this->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->Icon = (cli::safe\_cast<System::Drawing::Icon^>(resources->GetObject(L"$this.Icon")));

this->MaximumSize = System::Drawing::Size(1920, 1080);

this->MinimumSize = System::Drawing::Size(763, 675);

this->Name = L"LAB\_3";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Лаборатораня работа №3. Строки и работа с файлами";

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//

// Определение, чего больше в тексте, букв или цифр

//

private: void count()

{

int int\_q = 0, char\_q = 0;

if (this->text->Text->StartsWith("Цифры\r\n"))

{

this->text->Text = this->text->Text->Replace("Цифры\r\n", "");

}

else if (this->text->Text->StartsWith("Буквы\r\n"))

{

this->text->Text = this->text->Text->Replace("Буквы\r\n", "");

}

String^ str = this->text->Text;

for (int i = 0; i < str->Length; i++)

{

if (str[i] >= '0' && str[i] <= '9')

{

int\_q++;

}

else if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= 1040 && str[i] <= 1103))

{

char\_q++;

}

}

if (char\_q > int\_q)

{

this->text->Text = "Буквы\r\n" + text->Text;

}

else if (char\_q < int\_q)

{

this->text->Text = "Цифры\r\n" + text->Text;

}

}

//

// Вставка символа процента после каждого вхождения впоросительного знака

//

private: void insert\_percent(String^ str)

{

if (str == "?") str += "%";

else

{

for (int i = 0; i < str->Length - 1; i++)

{

if (str->Length > 1 && str[i] == '?' && str[i + 1] != '%')

{

str = str->Insert(i + 1, "%");

}

}

}

if (str->EndsWith("?")) str += "%";

this->text->Text = str;

}

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//

// Вывод имя файла и его расширения

//

private: void file\_info(String^ info)

{

this->extension->Text = info->Remove(0, info->Length - (info->Length - info->LastIndexOf(".")));

String^ temp = info->Remove(0, info->LastIndexOf("\\") + 1);

this->name\_of\_file->Text = temp->Remove(temp->Length - (temp->Length - temp->LastIndexOf(".")), 4);

}

//

// Чтение из файла

//

private: void read\_from\_file(String^ filename)

{

if (filename->Contains(".txt"))

{

StreamReader^ reader = gcnew StreamReader(filename);

this->text->Text = reader->ReadToEnd();

reader->Close();

}

else if (filename->Contains(".bin"))

{

BinaryReader^ reader = gcnew BinaryReader(File::Open(filename, FileMode::OpenOrCreate));

this->text->Text = reader->ReadString();

reader->Close();

}

}

//

// Запись в файл

//

private: void write\_to\_file(String^ filename)

{

if (filename->Contains(".txt"))

{

StreamWriter^ writer = gcnew StreamWriter(filename, false);

writer->Write(text->Text);

writer->Close();

}

else if (filename->Contains(".bin"))

{

BinaryWriter^ writer = gcnew BinaryWriter(File::Open(filename, FileMode::OpenOrCreate));

writer->Write(this->text->Text);

writer->Close();

}

}

//

// Создание файла

//

private: void create\_file(String^ filename)

{

if (filename->Contains(".txt"))

{

StreamWriter^ writer = gcnew StreamWriter(filename, false);

writer->Close();

}

else if (filename->Contains(".bin"))

{

BinaryWriter^ writer = gcnew BinaryWriter(File::Open(filename, FileMode::OpenOrCreate));

writer->Close();

}

}

//

// Удаление файла

//

private: void delete\_file(String^ filename)

{

FileInfo^ file = gcnew FileInfo(filename);

if (file->Exists)

{

file->Delete();

}

}

//

// Переименовать файл

//

private: void rename(String^ new\_name)

{

if (open\_file\_name != "")

{

int sl\_index = 0, dot\_index = 0;

sl\_index = open\_file\_name->LastIndexOf("\\");

dot\_index = open\_file\_name->IndexOf(".");

delete\_file(open\_file\_name);

String^ way = (open\_file\_name->Remove(sl\_index + 1, dot\_index - sl\_index - 1))->Insert(sl\_index + 1, new\_name);

write\_to\_file(way);

open\_file\_name = way;

file\_info(way);

}

}

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//

// Действия при выборе создания файла

//

private: System::Void создатьToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

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

{

this->text->Text = "";

create\_file(SaveFileDialog->FileName);

open\_file\_name = SaveFileDialog->FileName;

file\_info(SaveFileDialog->FileName);

}

}

//

// Действия при выборе открытия файла

//

private: System::Void открытьToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

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

{

this->text->Text = "";

open\_file\_name = OpenFileDialog->FileName;

this->text->Clear();

read\_from\_file(OpenFileDialog->FileName);

file\_info(open\_file\_name);

}

}

//

// Действия при выборе сохранения файла

//

private: System::Void сохранитьToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (open\_file\_name != "")

{

write\_to\_file(open\_file\_name);

}

else

{

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

{

write\_to\_file(SaveFileDialog->FileName);

open\_file\_name = SaveFileDialog->FileName;

file\_info(SaveFileDialog->FileName);

}

}

}

//

// Действия при выборе редактирования (% после ?)

//

private: System::Void toolStripMenuItem2\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (this->text->Text != "") insert\_percent(this->text->Text);

else

{

MessageBox::Show("Файл пуст", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

//

// Действия при выборе редактирования (больше букв или цифр)

//

private: System::Void большеБуквИлиЦифрToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (this->text->Text != "") count();

else

{

MessageBox::Show("Файл пуст", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

//

// Действия при выборе конвертации .txt в .bin

//

private: System::Void txtВbinToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (open\_file\_name != "" && open\_file\_name->Contains(".txt"))

{

BinaryWriter^ writer = gcnew BinaryWriter(File::Open(open\_file\_name->Replace(".txt", ".bin"), FileMode::OpenOrCreate));

writer->Write(this->text->Text);

writer->Close();

delete\_file(open\_file\_name);

open\_file\_name = open\_file\_name->Replace(".txt", ".bin");

file\_info(open\_file\_name);

}

else if (open\_file\_name != "" && open\_file\_name->Contains(".bin"))

{

String^ temp;

BinaryReader^ reader = gcnew BinaryReader(File::Open(open\_file\_name, FileMode::Open));

temp = reader->ReadString();

reader->Close();

write\_to\_file(open\_file\_name->Replace(".bin", ".txt"));

delete\_file(open\_file\_name);

open\_file\_name = open\_file\_name->Replace(".bin", ".txt");

file\_info(open\_file\_name);

}

else

{

MessageBox::Show("Файл не открыт", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

//

// Действия при выборе очистки полей

//

private: System::Void очиститьПолеToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

this->text->Text = "";

this->open\_file\_name = "";

this->name\_of\_file->Text = "";

this->extension->Text = "";

}

//

// Действия при выборе переименования

//

private: System::Void переименоватьToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (open\_file\_name != "")

{

int sl\_index = open\_file\_name->LastIndexOf("\\");

String^ name = open\_file\_name->Remove(0, sl\_index + 1);

int dot\_index = name->IndexOf(".");

name = name->Remove(dot\_index, name->Length - dot\_index);

int x = this->Left + (this->Width / 2) - 200;

int y = this->Top + (this->Height / 2) - 100;

String^ new\_name = ";";

int flag = 1;

while (max\_quantity(new\_name) || valid\_input(new\_name))

{

if ((max\_quantity(new\_name) || valid\_input(new\_name)) && flag == 2)

{

MessageBox::Show("Некрорректный ввод\r\n\r\nНе допускаются символы , . \\ / | : ; \* ? \" < > = + [ ] { }\r\nМаксимальная длина имени - 250 символов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

if (flag) flag = 2;

new\_name = Interaction::InputBox("Введите новое имя файла\r\n\r\nТекущее имя файла: " + name, "Переименовать", "", x, y);

}

if (new\_name != "") rename(new\_name);

}

else

{

MessageBox::Show("Файл для переименования не открыт", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

//

// Проверка количества символов в названии

//

private: bool max\_quantity(String^ str)

{

if (str->Length > 250)

return true;

return false;

}

//

// Проверка корректности введенных символов

//

private: bool valid\_input(String^ str)

{

if ((str->Contains(",")) || (str->Contains(".")) || (str->Contains("\\")) || (str->Contains("/")) || (str->Contains(":")) || (str->Contains("\*")) || (str->Contains("?")) || (str->Contains("|")) || (str->Contains("\"")) || (str->Contains("<")) || (str->Contains(">")) || (str->Contains(";")) || (str->Contains("=")) || (str->Contains("+")) || (str->Contains("[")) || (str->Contains("]")) || (str->Contains("{")) || (str->Contains("}")))

return true;

return false;

}

//

// Дейстивя при выборе сохранения в

//

private: System::Void сохранитьВФайлToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

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

{

String^ name = SaveFileDialog->FileName;

write\_to\_file(name);

}

}

//

// Действия при выборе О программе

//

private: System::Void оПрограммеToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

MessageBox::Show("Программа для работы с текстовыми файлами и обработки их содержимого.\r\n\r\nВ программе предусмотрена возможность:\r\n1. Чтения определенного файла;\r\n2. Редактирования содержимого файла;\r\n3. Сохранение отредактированной информации в файл;\r\n4. Создание нового файла;\r\n5. Выбора файлов для использования;\r\n6. Конвертация из формата txt в bin, и обратно;\r\n7. Переименование открытого файла.\r\n\r\nПредусмотренные варианты редактирования содержимого:\r\n1. После каждого знака вопроса поставить без пробелов символ «%»;\r\n2. Подсчитать количество букв в тексте и сравнить его с количеством цифр в тексте, написать в начало текста «Буквы», если букв больше, «Цифры» в ином случае.", "О программе", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//

// Функция тестирования процедуры count(), на входе void, на выходе int

//

private: int count\_test()

{

this->text->Text = "abc11";

count();

if (this->text->Text->StartsWith("Буквы"))

{

this->text->Text = "";

return 0;

}

else

{

this->text->Text = "";

return 1;

}

}

//

// Функция тестирования процедуры insetr\_percent(), на входе void, на выходе int

//

private: int insert\_test()

{

String^ temp = "a?b?c";

insert\_percent(temp);

if(this->text->Text == "a?%b?%c")

{

this->text->Text = "";

return 0;

}

else

{

this->text->Text = "";

return 1;

}

}

//

// Функция тестирования процедуры file\_info(), на входе void, на выходе int

//

private: int file\_info\_test()

{

String^ file\_name = "C:\\Users\\name.txt";

file\_info(file\_name);

if(this->name\_of\_file->Text == "name" && this->extension->Text == ".txt")

{

this->name\_of\_file->Text = "";

this->extension->Text = "";

return 0;

}

else

{

this->name\_of\_file->Text = "";

this->extension->Text = "";

return 1;

}

}

//

// Действия при выборе Тест

//

private: System::Void тестToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

int result = count\_test() + insert\_test() + file\_info\_test();

if (result == 0)

{

MessageBox::Show("Тесты пройдены, приложение работает корректно", "Тестирование", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

else

{

MessageBox::Show("Тесты не пройдены. Приложение будет закрыто", "Тестирование", MessageBoxButtons::OK, MessageBoxIcon::Error);

this->Close();

}

}

//

// Процедура тестирования программы при вызове конструктора

//

private: void constructor\_test()

{

int result = count\_test() + insert\_test() + file\_info\_test();

if (result != 0)

{

MessageBox::Show("Тесты не пройдены. Приложение будет закрыто", "Тестирование", MessageBoxButtons::OK, MessageBoxIcon::Error);

this->Close();

}

}

};

}

Приложение И

Листинг файла LAB\_3.cpp

#include "LAB\_3.h"

#include "Windows.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute]

int main()

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

LAB3::LAB\_3 form;

Application::Run(% form);

}

Приложение И.

Листинг файла UnitTestLAB\_3.cpp

#include "pch.h"

#include "CppUnitTest.h"

#include "..\LAB\_3\Functions.h"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTestLAB3

{

TEST\_CLASS(UnitTestLAB3)

{

public:

TEST\_METHOD(count\_test)

{

string test = "abc11";

Assert::IsTrue(count(&test) == "Буквы\nabc11");

}

TEST\_METHOD(insert\_test)

{

string test = "a?b?c";

Assert::IsTrue(insert\_percent(&test) == "a?%b?%c");

}

TEST\_METHOD(file\_info\_test)

{

string file\_name = "C:\\Users\\name.txt", extension = "", name\_of\_file = "";

file\_info(file\_name, &extension, &name\_of\_file);

Assert::IsTrue(extension == ".txt" && name\_of\_file == "name");

}

};

}

Приложение К.

Листинг файла Functions.h

#pragma once

#include <string>

using namespace std;

string count(string \*text)

{

int int\_q = 0, char\_q = 0;

string str = \*text;

for (int i = 0; i < str.length(); i++)

{

if (str[i] >= '0' && str[i] <= '9')

{

int\_q++;

}

if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= 1040 && str[i] <= 1103))

{

char\_q++;

}

}

if (char\_q > int\_q)

{

\*text = "Буквы\n" + \*text;

}

else if (char\_q < int\_q)

{

\*text = "Цифры\n" + \*text;

}

return \*text;

}

string insert\_percent(string\* input)

{

string str = \*input;

if (str == "?") str += "%";

else

{

for (int i = 0; i < str.length() - 1; i++)

{

if (str.length() > 1 && str[i] == '?' && str[i + 1] != '%')

{

str = str.insert(i + 1, "%");

}

}

}

if (str[str.length() - 1] == '?') str += " % ";

\*input = str;

return \*input;

}

int last\_indx\_of(string str, char sym)

{

int indx = 0;

for (int i = 0; i < str.length(); i++)

{

if (str[i] == sym) indx = i;

}

return indx;

}

void file\_info(string info, string\* extension, string\* name\_of\_file)

{

int dot = last\_indx\_of(info, '.');

int slash = last\_indx\_of(info, '\\');

string temp = info;

\*extension = temp.erase(0, temp.length() - (temp.length() - dot));

temp = info.erase(0, slash + 1);

dot = last\_indx\_of(temp, '.');

temp = temp.erase(temp.length() - (temp.length() - dot), 4);

\*name\_of\_file = temp;

}