МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения

вычислительной техники и автоматизированных

систем

**Лабораторная работа №1**

по дисциплине: ООП

тема: «Знакомство с интегрированной средой разработки (ИСР) Microsoft Visual Studio или QT»

Выполнил: студент группы ВТ-231

Самойленко Кирилл

Проверили:  
Буханов Д. Г.

Белгород 2025

**Цель работы**:изучение функциональных возможностей интегрированной среды

разра-ботки (ИСР) Visual Studio или QT.

**Задания**:

**Вариант** **17**

1.B текстовом файле несколько последовательных одинаковых символов заменить одним.

2.Дан файл, компоненты которого являются целыми числами. Преобразовать его, исключив повторные вхождения одного и того же числаЭти задачи в схожи что у нас подаются текстовые файлы,и в одном нужно заменитьнесколько символов одним в во второй задачи просто убрать повторение подряд идущих символов.  
  
**Задание 1:**

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <algorithm>

#include <locale.h>

using namespace std;

string removeRepeatedChars(const string& text) {

if (text.empty()) {

return "";

}

string result;

result += text[0];

for (size\_t i = 1; i < text.length(); ++i) {

if (text[i] != text[i - 1]) {

result += text[i];

}

}

return result;

}

void removeRepeatedNumbers(const string& inputFilename, const string& outputFilename) {

ifstream inputFile(inputFilename);

if (!inputFile.is\_open()) {

cerr << "Ошибка: Не удалось открыть файл " << inputFilename << endl;

return;

}

vector<int> numbers;

int number;

while (inputFile >> number) {

numbers.push\_back(number);

}

inputFile.close();

vector<int> uniqueNumbers;

for (int num : numbers) {

if (find(uniqueNumbers.begin(), uniqueNumbers.end(), num) == uniqueNumbers.end()) {

uniqueNumbers.push\_back(num);

}

}

ofstream outputFile(outputFilename);

if (!outputFile.is\_open()) {

cerr << "Ошибка: Не удалось открыть файл " << outputFilename << " для записи" << endl;

return;

}

for (int num : uniqueNumbers) {

outputFile << num << " ";

}

outputFile.close();

cout << "Файл " << inputFilename << " успешно обработан. Результат записан в "

<< outputFilename << endl;

}

int main() {

setlocale(LC\_ALL, "Rus");

int choice;

cout << "Выберите действие:" << endl;

cout << "1. Замена последовательных одинаковых символов в текстовом файле одним." << endl;

cout << "2. Удаление повторяющихся целых чисел из файла." << endl;

cout << "Ваш выбор: ";

cin >> choice;

cin.ignore();

if (choice == 1) {

string inputFilename = "C:\\Users\\Assa\\input1.txt";

string outputFilename = "C:\\Users\\Assa\\output1.txt";

ifstream inputFile(inputFilename);

if (!inputFile.is\_open()) {

cerr << "Ошибка: Не удалось открыть файл " << inputFilename << endl;

return 1;

}

string line;

string content;

while (getline(inputFile, line)) {

content += line + "\n";

}

inputFile.close();

string processedContent = removeRepeatedChars(content);

ofstream outputFile(outputFilename);

if (!outputFile.is\_open()) {

cerr << "Ошибка: Не удалось открыть файл " << outputFilename << " для записи" << endl;

return 1;

}

outputFile << processedContent;

outputFile.close();

cout << "Файл " << inputFilename << " успешно обработан. Результат записан в "

<< outputFilename << "." << endl;

}

else if (choice == 2) {

string inputFilename = "C:\\Users\\Assa\\input2.txt";

string outputFilename = "C:\\Users\\Assa\\output2.txt";

removeRepeatedNumbers(inputFilename, outputFilename);

}

else {

cout << "Неверный выбор." << endl;

return 1;

}

return 0;

} **Задание 2:**

#pragma once

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::Collections::Generic;

namespace excel {

using namespace System;

using namespace System::Windows::Forms;

public ref class MyForm : public System::Windows::Forms::Form {

public:

MyForm(void) {

InitializeComponent();

InitializeRibbonCustomization();

}

protected:

~MyForm() {

if (components) {

delete components;

}

}

private:

System::Windows::Forms::ComboBox^ chooseCommandComboBox;

System::Windows::Forms::ListBox^ availableCommandsListBox;

System::Windows::Forms::TreeView^ ribbonTreeView;

System::Windows::Forms::Button^ addButton;

System::Windows::Forms::Button^ removeButton;

System::Windows::Forms::Button^ newTabButton;

System::Windows::Forms::Button^ newGroupButton;

System::Windows::Forms::Button^ renameButton;

System::Windows::Forms::Button^ resetButton;

System::Windows::Forms::Button^ importExportButton;

System::Windows::Forms::Button^ okButton;

System::Windows::Forms::Button^ cancelButton;

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

System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code

void InitializeComponent(void) {

// Код инициализации компонентов

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

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

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

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

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

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

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

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

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

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

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

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

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

this->SuspendLayout();

// Настройка компонентов chooseCommandComboBox

this->chooseCommandComboBox->FormattingEnabled = true;

this->chooseCommandComboBox->Items->AddRange(gcnew cli::array<System::Object^>(2) {

L"Часто используемые команды", L"Все команды"

});

this->chooseCommandComboBox->Location = System::Drawing::Point(252, 23);

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

this->chooseCommandComboBox->Size = System::Drawing::Size(252, 21);

this->chooseCommandComboBox->TabIndex = 0;

// Настройка availableCommandsListBox

this->availableCommandsListBox->FormattingEnabled = true;

this->availableCommandsListBox->Items->AddRange(gcnew cli::array<System::Object^>(16) {

L"Быстрая печать", L"Вернуть", L"Все типы диаграмм...",

L"Вставить", L"Вставить рисунок", L"Вставить столбцы на лист",

L"Вставить строки на лист", L"Вставить таблицу",

L"Вставить функцию...", L"Вставить ячейки...", L"Вырезать",

L"Выровнять по центру", L"Границы", L"Диспетчер имен",

L"Добавить или удалить фильтры"

});

this->availableCommandsListBox->Location = System::Drawing::Point(12, 59);

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

this->availableCommandsListBox->Size = System::Drawing::Size(234, 342);

this->availableCommandsListBox->TabIndex = 1;

// Настройка ribbonTreeView

this->ribbonTreeView->CheckBoxes = true;

this->ribbonTreeView->Location = System::Drawing::Point(252, 59);

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

this->ribbonTreeView->Size = System::Drawing::Size(340, 342);

this->ribbonTreeView->TabIndex = 2;

// Настройка кнопок

this->addButton->Location = System::Drawing::Point(12, 407);

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

this->addButton->Size = System::Drawing::Size(111, 23);

this->addButton->TabIndex = 3;

this->addButton->Text = L"Добавить >>";

this->addButton->UseVisualStyleBackColor = true;

// ... (аналогично для других кнопок)

// Завершение настройки формы

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

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

this->ClientSize = System::Drawing::Size(604, 514);

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

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

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

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

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

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

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

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

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

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

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

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

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

this->Name = L"MyForm";

this->Text = L"Параметры Excel";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

void InitializeRibbonCustomization() {

// Установка значения по умолчанию для комбобокса

chooseCommandComboBox->SelectedIndex = 0;

// Создание иерархии вкладок Excel

TreeNode^ rootNode = gcnew TreeNode("Основные вкладки");

ribbonTreeView->Nodes->Add(rootNode);

// Добавление узла "Удаление фона"

TreeNode^ removeBackgroundNode = gcnew TreeNode("Удаление фона");

removeBackgroundNode->Checked = true;

rootNode->Nodes->Add(removeBackgroundNode);

// Вкладка "Главная" с подкатегориями

TreeNode^ homeTabNode = gcnew TreeNode("Главная");

homeTabNode->Checked = true;

rootNode->Nodes->Add(homeTabNode);

// Подкатегории вкладки "Главная"

homeTabNode->Nodes->Add(gcnew TreeNode("Буфер обмена"));

homeTabNode->Nodes->Add(gcnew TreeNode("Шрифт"));

homeTabNode->Nodes->Add(gcnew TreeNode("Выравнивание"));

homeTabNode->Nodes->Add(gcnew TreeNode("Число"));

homeTabNode->Nodes->Add(gcnew TreeNode("Стили"));

homeTabNode->Nodes->Add(gcnew TreeNode("Ячейки"));

homeTabNode->Nodes->Add(gcnew TreeNode("Редактирование"));

// Добавление других вкладок

TreeNode^ insertTabNode = gcnew TreeNode("Вставка");

insertTabNode->Checked = true;

rootNode->Nodes->Add(insertTabNode);

insertTabNode->Nodes->Add(gcnew TreeNode("Рисование"));

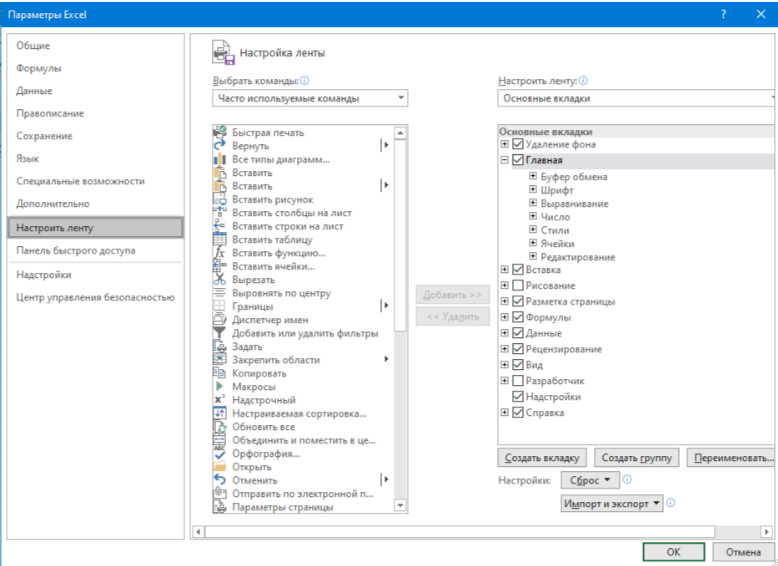
// ... (аналогично для других вкладок)

// Раскрытие всех узлов

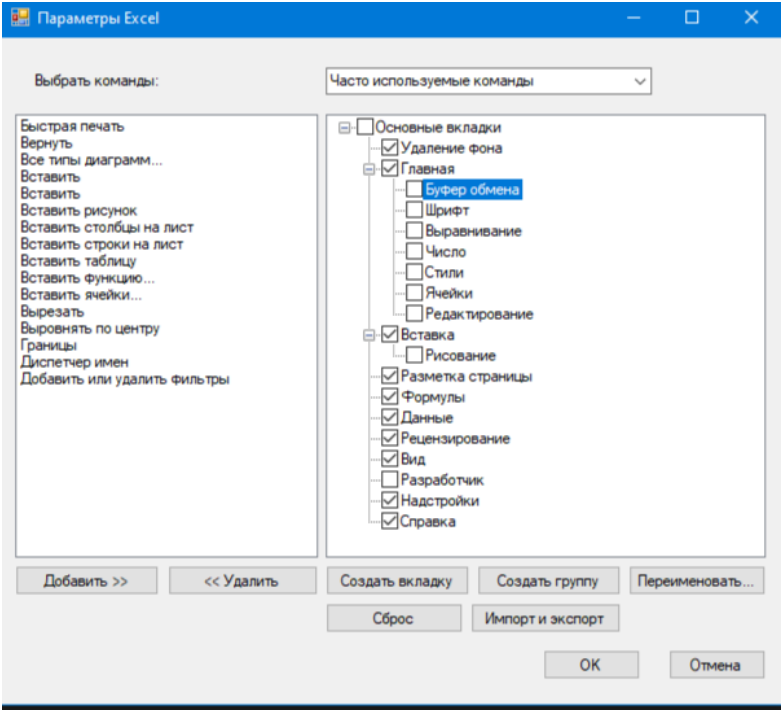
rootNode->ExpandAll();

}

};

}

exel

****

результат работы программы