

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: "Объектно-ориентированное программирование"

по теме: "Знакомство с интегрированной средой разработки (ИСР) Microsoft Visual  
Studio или QT"

Выполнил: ст. группы КБ-231  
Локтионов Станислав Александрович

Проверил:  
Морозов Данил Александрович

**Цель работы:**изучение функциональных возможностей интегрированной среды разработки (ИСР) Visual Studio или QT.

## Вариант 17

Решение задачи А и В:

1.В 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 числа

Эти задачи схожи тем что у нас подаются текстовые файлы,и в одном нужно заменить несколько символов одним а во второй задачи просто убрать повторение подряд идущих символов.

Для решения 1 задачи воспользуемся таким алгоритмом:

- Откроем файл input1.txt(если он не пуст).
- Считываем символы, если текущий не равен предыдущему,записываем в результат.
- Возвращаем результат.
- Открываем файл output1.txt.
- После записываем в готовый файл output.txt.

Для решения второй задачи предлагается следующий алгоритм:

- 1) Открыть файл input2.txt.
- 2) Если файл не пуст, считать данные из файла, интерпретируя их как вектор чисел.
- 3) Проверить наличие уникальных элементов в считанном векторе.
- 4) Если в векторе есть уникальные элементы, добавить их в файл output2.txt.
- 5) Открыть файл output2.txt для записи (добавления).
- 6) Записать уникальные элементы в файл output2.txt.

Код программы:

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <algorithm>
#include <locale.h>

using namespace std;

// Функция для удаления повторяющихся символов в строке
string removeRepeatedChars(const string& text) {
    if (text.empty()) {
```

```

        return "";
    }

    string result;
    result += text[0]; // Добавляем первый символ

    for (size_t i = 1; i < text.length(); ++i) {
        if (text[i] != text[i - 1]) { // Если текущий символ не равен предыдущему
            result += text[i];
        }
    }

    return result;
}

// Функция для удаления повторяющихся чисел из файла
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) { // Чтение чисел из файла
        numbers.push_back(number);
    }
    inputFile.close();

    // Удаляем повторяющиеся числа, сохраняя порядок
    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 <<
        cerr << " для записи" << endl;
        return;
    }

    for (int num : uniqueNumbers) {
        outputFile << num << " "; // Запись уникальных чисел в файл
    }
}

```

```

outputFile.close();

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

int main() {
    setlocale(LC_ALL, "Rus");
    int choice;

    cout << "Выберите действие:" << endl;
    cout << "1. Замена последовательных одинаковых символов в << endl
    cout <<"текстовом файле одним." << endl;
    cout << "2. Удаление повторяющихся целых чисел из файла." << endl;
    cout << "Ваш выбор: ";
    cin >> choice;
    cin.ignore(); // Очистка буфера ввода после cin >> choice

    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"; // Сохраняем структуру файла с переносами строк
        }
        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; // Записываем обработанный текст
        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;
}

```

После выполнения программы исходные файлы преобразуются по условиям задач А и В.

**Задание 2** Пошаговое описание создания Windows Forms в Visual Studio 2022:

- 1) Открываем Visual Studio 2022 и создаем новый проект C++/CLI с использованием Windows Forms.
- 2) Изменяем свойства проекта:
  - В разделе "Подсистема" указываем **Windows (/SUBSYSTEM:WINDOWS)**.
  - В разделе "Дополнительно" в поле "Точка входа" указываем **main**.
- 3) Создаем файлы формата .h и .cpp.
- 4) Описываем код в файле формата .h.
- 5) Основные компоненты кода:

**Пространство имен:**

```

namespace excel {
// ...
}

```

Пространство имен **excel** используется для организации кода и предотвращения конфликтов имен с другими частями программы или библиотеками.

**Класс MyForm:**

```

public ref class MyForm : public System::Windows::Forms::Form {
// ...
};

```

Класс **MyForm** наследует от **System::Windows::Forms::Form**, что делает его формой (окном) в приложении Windows. Внутри класса определяются все элементы управления и логика работы.

**Конструктор:**

```

MyForm(void)
{
    InitializeComponent();
    InitializeRibbonCustomization(); // Вызов функции инициализации
}

```

Конструктор инициализирует компоненты формы и вызывает метод `InitializeRibbonCustomization()`, который заполняет элементы управления начальными данными.

#### Метод `InitializeComponent`:

Этот метод автоматически сгенерирован дизайнером и отвечает за создание и настройку всех элементов управления на форме. Здесь определяются такие элементы, как:

- `ComboBox (chooseCommandComboBox)`: выпадающий список для выбора типа команд.
- `ListBox (availableCommandsListBox)`: список доступных команд, которые можно добавить в ленту.
- `TreeView (ribbonTreeView)`: дерево, представляющее структуру командной ленты Excel, где пользователи могут видеть и управлять вкладками и командами.
- Кнопки: различные кнопки для выполнения действий, таких как добавление команд, удаление, создание новых вкладок и групп, переименование и сброс настроек.

#### Метод `InitializeRibbonCustomization`:

Этот метод заполняет `TreeView` начальными данными, представляющими различные вкладки и команды Excel. Он создает иерархическую структуру, где каждая вкладка может содержать команды и подкатегории. Например, вкладка "Главная" содержит команды для работы со шрифтами, выравниванием и т.д. Это позволяет пользователю видеть, какие команды доступны и как они организованы.

Код программы:

```

#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;

```

```

    /// <summary>
    /// Summary for MyForm
    /// </summary>
public ref class MyForm : public System::Windows::Forms::Form
{
public:
    MyForm(void)
    {
        InitializeComponent();
        //
        //TODO: Add the constructor code here
        //
        InitializeRibbonCustomization(); // Call the initialization
        ↪ function
    }

protected:
    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    ~MyForm()
    {
        if (components)
        {
            delete components;
        }
    }

private: System::Windows::Forms::ComboBox^ chooseCommandComboBox;
private: System::Windows::Forms::ListBox^ availableCommandsListBox;
private: System::Windows::Forms::TreeView^ ribbonTreeView;
private: System::Windows::Forms::Button^ addButton;
private: System::Windows::Forms::Button^ removeButton;
private: System::Windows::Forms::Button^ newTabButton;
private: System::Windows::Forms::Button^ newGroupButton;
private: System::Windows::Forms::Button^ renameButton;
private: System::Windows::Forms::Button^ resetButton;
private: System::Windows::Forms::Button^ importExportButton;
private: System::Windows::Forms::Button^ okButton;
private: System::Windows::Forms::Button^ cancelButton;
private: System::Windows::Forms::Label^ label1;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container^ components;
#pragma region Windows Form Designer generated code
    /// <summary>
    /// Required method for Designer support - do not modify

```

```

    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        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();

        //
        // 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;

        //
        // 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", 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;

        //
        // 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;
//
// addButton
//
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;
//
// removeButton
//
this->removeButton->Location = System::Drawing::Point(130, 407);
this->removeButton->Name = L"removeButton";
this->removeButton->Size = System::Drawing::Size(116, 23);
this->removeButton->TabIndex = 4;
this->removeButton->Text = L"<< Удалить";
this->removeButton->UseVisualStyleBackColor = true;
//
// newTabButton
//
this->newTabButton->Location = System::Drawing::Point(252, 407);
this->newTabButton->Name = L"newTabButton";
this->newTabButton->Size = System::Drawing::Size(111, 23);
this->newTabButton->TabIndex = 5;
this->newTabButton->Text = L"Создать вкладку";
this->newTabButton->UseVisualStyleBackColor = true;
//
// newGroupButton
//
this->newGroupButton->Location = System::Drawing::Point(369,
↵ 407);
this->newGroupButton->Name = L"newGroupButton";
this->newGroupButton->Size = System::Drawing::Size(111, 23);
this->newGroupButton->TabIndex = 6;
this->newGroupButton->Text = L"Создать группу";
this->newGroupButton->UseVisualStyleBackColor = true;
//
// renameButton
//
this->renameButton->Location = System::Drawing::Point(486, 407);
this->renameButton->Name = L"renameButton";
this->renameButton->Size = System::Drawing::Size(106, 23);
this->renameButton->TabIndex = 7;

```

```

this->renameButton->Text = L"Переименовать...";
this->renameButton->UseVisualStyleBackColor = true;
//
// resetButton
//
this->resetButton->Location = System::Drawing::Point(252, 436);
this->resetButton->Name = L"resetButton";
this->resetButton->Size = System::Drawing::Size(106, 23);
this->resetButton->TabIndex = 8;
this->resetButton->Text = L"Сброс";
this->resetButton->UseVisualStyleBackColor = true;
//
// importExportButton
//
this->importExportButton->Location = System::Drawing::Point(364,
↪ 436);
this->importExportButton->Name = L"importExportButton";
this->importExportButton->Size = System::Drawing::Size(116, 23);
this->importExportButton->TabIndex = 9;
this->importExportButton->Text = L"Импорт и экспорт";
this->importExportButton->UseVisualStyleBackColor = true;
//
// okButton
//
this->okButton->Location = System::Drawing::Point(420, 472);
this->okButton->Name = L"okButton";
this->okButton->Size = System::Drawing::Size(75, 23);
this->okButton->TabIndex = 10;
this->okButton->Text = L"OK";
this->okButton->UseVisualStyleBackColor = true;
//
// cancelButton
//
this->cancelButton->Location = System::Drawing::Point(517, 472);
this->cancelButton->Name = L"cancelButton";
this->cancelButton->Size = System::Drawing::Size(75, 23);
this->cancelButton->TabIndex = 11;
this->cancelButton->Text = L"Отмена";
this->cancelButton->UseVisualStyleBackColor = true;
//
// label1
//
this->label1->AutoSize = true;
this->label1->Location = System::Drawing::Point(24, 26);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(97, 13);
this->label1->TabIndex = 12;
this->label1->Text = L"Выбрать команды:";
//
// MyForm

```

```

//
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

private:
    void InitializeRibbonCustomization()
    {
        // Установить значение по умолчанию для комбобокса
        chooseCommandComboBox->SelectedIndex = 0;

        // Заполнить ribbonTreeView тестовыми данными (замените на реальные данные)
        TreeNode^ rootNode = gcnew TreeNode("Основные вкладки");
        ribbonTreeView->Nodes->Add(rootNode);

        TreeNode^ removeBackgroundNode = gcnew TreeNode("Удаление фона");
        removeBackgroundNode->Checked = true; // Установить "галочку" в этом
        ↪ узле
        rootNode->Nodes->Add(removeBackgroundNode);

        TreeNode^ homeTabNode = gcnew TreeNode("Главная");
        homeTabNode->Checked = true; // Установить "галочку" для вкладки
        ↪ "Главная"
        rootNode->Nodes->Add(homeTabNode);

        TreeNode^ clipboardNode = gcnew TreeNode("Буфер обмена");
        homeTabNode->Nodes->Add(clipboardNode); // Добавить узел "Буфер
        ↪ обмена" в "Главная"
    }

```

```

TreeNode^ fontNode = gcnew TreeNode("Шрифт");
homeTabNode->Nodes->Add(fontNode); // Добавить узел "Шрифт" в
↳ "Главная"

TreeNode^ alignmentNode = gcnew TreeNode("Выравнивание");
homeTabNode->Nodes->Add(alignmentNode); // Добавить узел
↳ "Выравнивание" в "Главная"

TreeNode^ numberNode = gcnew TreeNode("Число");
homeTabNode->Nodes->Add(numberNode); // Добавить узел "Число" в
↳ "Главная"

TreeNode^ stylesNode = gcnew TreeNode("Стили");
homeTabNode->Nodes->Add(stylesNode); // Добавить узел "Стили" в
↳ "Главная"

TreeNode^ cellsNode = gcnew TreeNode("Ячейки");
homeTabNode->Nodes->Add(cellsNode); // Добавить узел "Ячейки" в
↳ "Главная"

TreeNode^ editingNode = gcnew TreeNode("Редактирование");
homeTabNode->Nodes->Add(editingNode); // Добавить узел
↳ "Редактирование" в "Главная"

TreeNode^ insertTabNode = gcnew TreeNode("Вставка");
insertTabNode->Checked = true; // Установить "галочку" для вкладки
↳ "Вставка"
rootNode->Nodes->Add(insertTabNode);

TreeNode^ illustrationsNode = gcnew TreeNode("Рисование");
insertTabNode->Nodes->Add(illustrationsNode); // Добавить узел
↳ "Рисование" в "Вставка"

TreeNode^ pageLayoutTabNode = gcnew TreeNode("Разметка страницы");
pageLayoutTabNode->Checked = true; // Установить "галочку" для вкладки
↳ "Разметка страницы"
rootNode->Nodes->Add(pageLayoutTabNode);

TreeNode^ formulasTabNode = gcnew TreeNode("Формулы");
formulasTabNode->Checked = true; // Установить "галочку" для вкладки
↳ "Формулы"
rootNode->Nodes->Add(formulasTabNode);

TreeNode^ dataTabNode = gcnew TreeNode("Данные");
dataTabNode->Checked = true; // Установить "галочку" для вкладки "Данные"
rootNode->Nodes->Add(dataTabNode);

TreeNode^ reviewTabNode = gcnew TreeNode("Рецензирование");
reviewTabNode->Checked = true; // Установить "галочку" для вкладки
↳ "Рецензирование"

```

```

rootNode->Nodes->Add(reviewTabNode);

TreeNode^ viewTabNode = gcnew TreeNode("Вид");
viewTabNode->Checked = true; // Установить "галочку" для вкладки "Вид"
rootNode->Nodes->Add(viewTabNode);

TreeNode^ developerTabNode = gcnew TreeNode("Разработчик");
developerTabNode->Checked = false; // Установить "галочку" для вкладки
↳ "Разработчик" (не отмечено)
rootNode->Nodes->Add(developerTabNode);

TreeNode^ addInsTabNode = gcnew TreeNode("Надстройки");
addInsTabNode->Checked = true; // Установить "галочку" для вкладки
↳ "Надстройки"
rootNode->Nodes->Add(addInsTabNode);

TreeNode^ helpTabNode = gcnew TreeNode("Справка");
helpTabNode->Checked = true; // Установить "галочку" для вкладки "Справка"
rootNode->Nodes->Add(helpTabNode);

        // Раскрыть все узлы, чтобы показать структуру
rootNode->ExpandAll();
    }
};
}

```

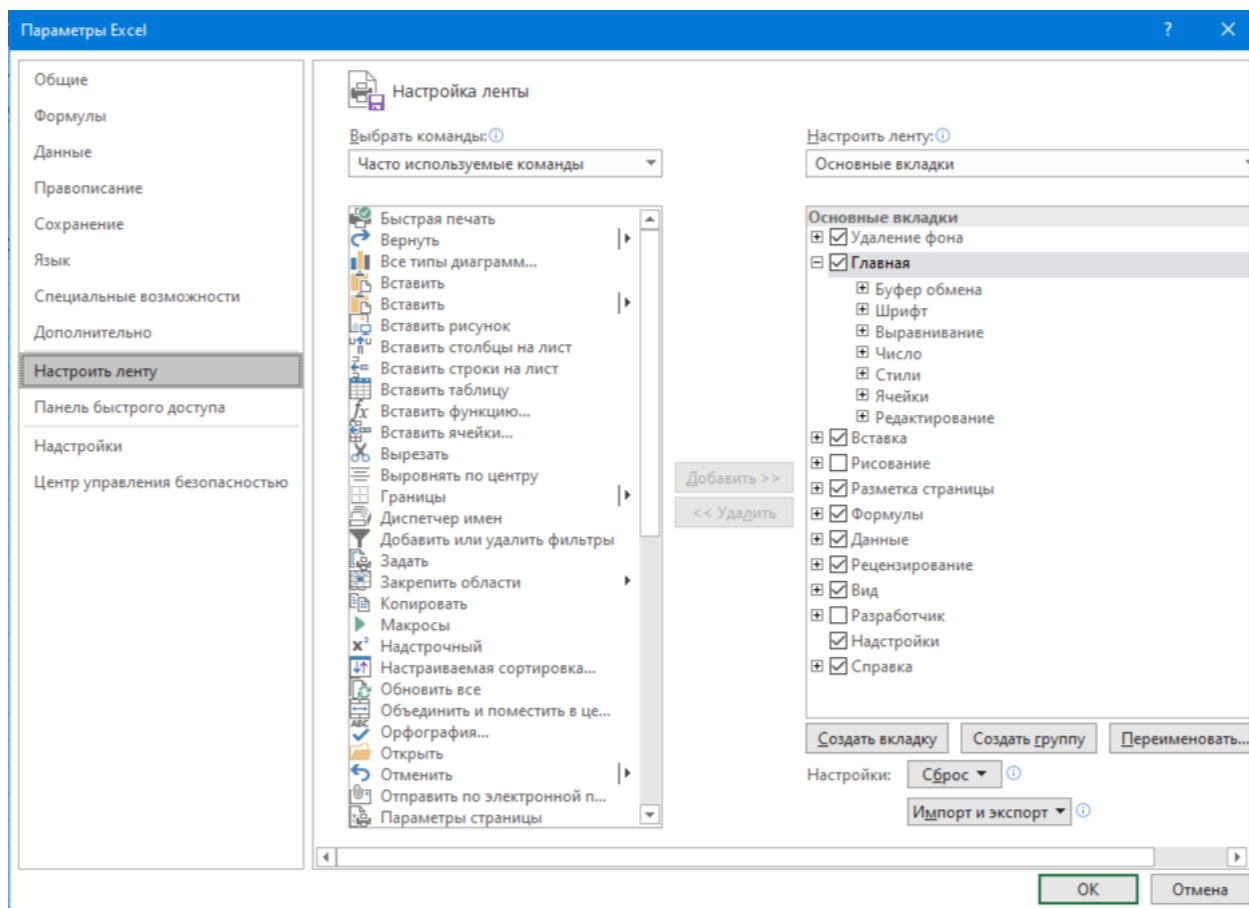


Рис. 1. Excel

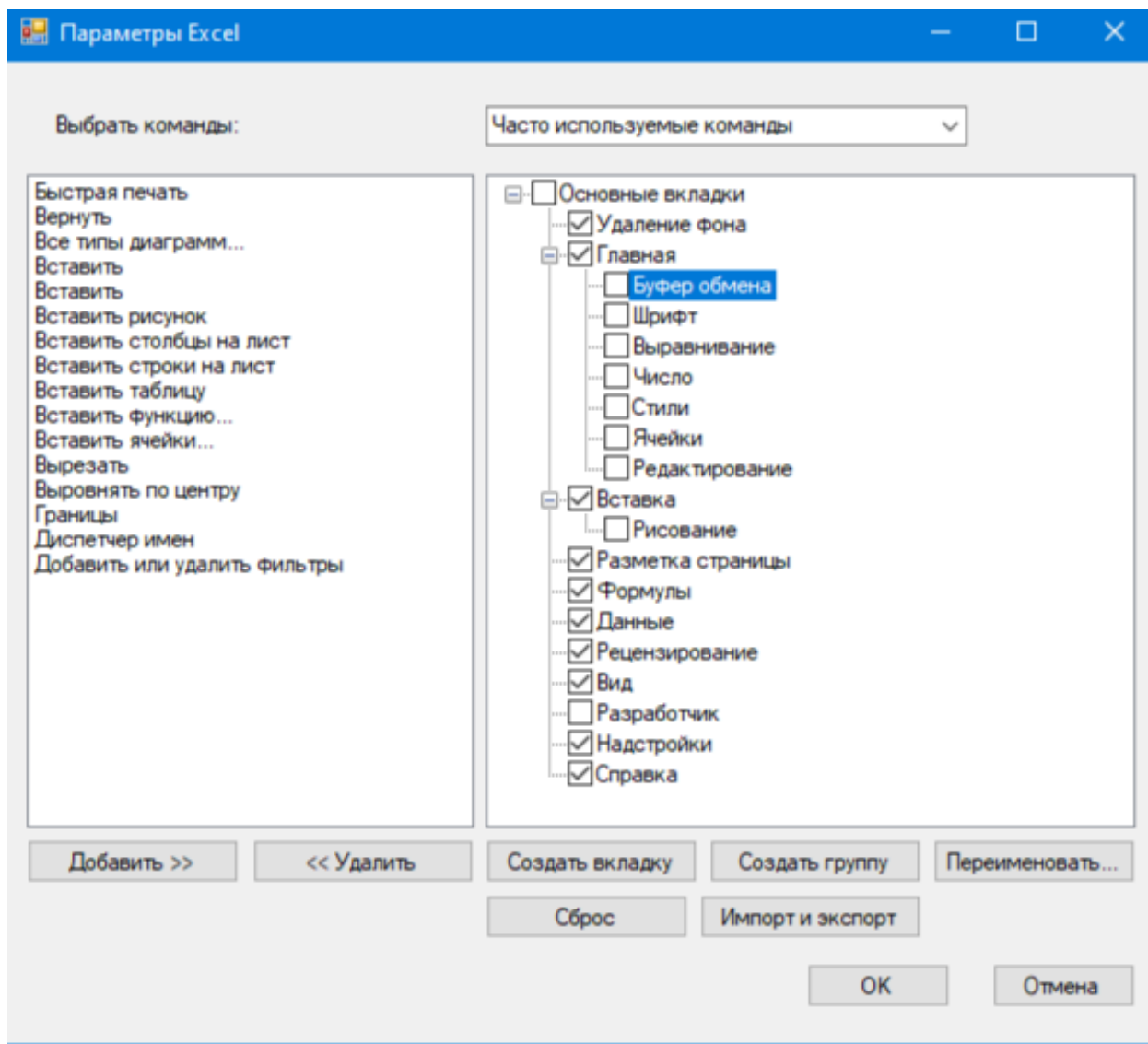


Рис. 2. Вывод программы

Вывод:изучили функциональных возможностей интегрированной среды разра-ботки (ИСП) Visual Studio или QT.