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

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

Выполнил: ст. группы ПВ-233

Ситников Алексей Павлович

Проверил:

Белгород 2025 г.

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

Вариант 13

Для того чтобы создать консольное приложение в Visual Studio нужно открыть приложение, нажать на файл, затем создать проект, выбрать «консольное приложение», нажать создать.

**Задание a. Дан текстовый файл. Преобразовать его, оставив в каждой строке только самое длинное слово.**

Решение:

#include <iostream>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

int main() {

ifstream f;

f.open("file.txt");

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

cout << "is not open";

return 0;

}

vector<string> words;

string line;

while (getline(f, line)) {

int max = -1;

int count = 0;

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

if (line[i] != ' ') {

count++;

}

else {

max = max > count ? max : count;

count = 0;

}

}

max = max > count ? max : count;

count = 0;

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

if (line[i] != ' ') {

count++;

}

else {

count = 0;

}

if (count == max) {

string word = line.substr(i-max + 1, max);

words.push\_back(word);

break;

}

}

}

f.clear();

f.close();

ofstream fin("file.txt");

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

cout << "is not open";

return 0;

}

for (int i = 0; i < words.size(); i++) {

fin << words[i];

fin << '\n';

}

f.close();

cout << endl;

system("pause");

return 1;

}

Тестовые данные:  
изначальный вид файла:  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Конечный вид файла:  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Задание б. B файле хранятся последовательности целых чисел, отличных от нуля. Ноль разделитель последовательностей. Преобразовать файл, сохранив в каждой последовательности числа между первым и вторым отрицательным числом. Если отрицательных чисел меньше двух, последовательность исключить из файла.**

#include <iostream>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

int main() {

ifstream f;

f.open("ff.txt");

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

cout << "is not open";

return 0;

}

vector<string> letters;

string line;

bool between\_negetiv = false;

bool is\_useless = true;

int count = 0;

int count\_negetiv = 0;

while(getline(f, line)) {

if ((char)line[0] != '0' && is\_useless) {

if ((char)line[0] == '-') {

between\_negetiv = !between\_negetiv;

is\_useless = between\_negetiv;

count\_negetiv++;

if (!is\_useless) {

if(count!=0)

letters.push\_back("0\n");

count++;

}

}

else if (between\_negetiv) {

letters.push\_back(line);

letters.push\_back("\n");

count+=2;

}

}

else if (count\_negetiv == 1) {

for (int i = 0; i < count; i++) {

letters.pop\_back();

}

count = 0;

count\_negetiv = 0;

}

if (line[0] == '0') {

between\_negetiv = false;

is\_useless = true;

count = 0;

count\_negetiv = 0;

}

}

f.clear();

f.close();

ofstream fin("ff.txt");

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

cout << "is not open";

return 0;

}

for (int i = 0; i < letters.size(); i++) {

fin << letters[i];

}

f.close();

cout << endl;

system("pause");

return 1;

}

Тестовые данные:  
Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Конечный вид файла:  
Изображение выглядит как текст, снимок экрана, Шрифт, типография

Контент, сгенерированный ИИ, может содержать ошибки.

Для создания экранной формы нужно создать пустой проект CLR, затем добавить форму Windows Forms. Затем на панели «вид» открываем панель элементов, из списка элементов выбираем нужные под проект. Затем в файле .cpp пишем:  
#include "MyForm.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThread]

int main(array<String^>^ args)

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

<имя проекта>::MyForm form;

Application::Run(% form);

return 1;

затем в файле myForm.h работаем над формой.

**Задание: Форма «Параметры Word» MS Word, вкладка «Правописание»**

Исходный код:

#pragma once

#include <iostream>

namespace Project4 {

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 MyForm

/// </summary>

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

{

public:

MyForm(void)

{

ReadFile();

InitializeComponent();

}

private:

String^ line;

void ReadFile()

{

try

{

StreamReader^ reader = gcnew StreamReader("file.txt");

line = reader->ReadLine();

reader->Close(); // Закрытие reader

}

catch (FileNotFoundException^ ex)

{

Console::WriteLine("File not found: " + ex->Message);

}

catch (Exception^ ex)

{

Console::WriteLine("An error occurred: " + ex->Message);

}

}

void SaveFile()

{

try

{

StreamWriter^ writer = gcnew StreamWriter("file.txt", false); // false означает перезапись файла

writer->WriteLine(line); // Запись строки в файл

writer->Close(); // Закрытие writer

}

catch (Exception^ ex)

{

Console::WriteLine("An error occurred while saving the file: " + ex->Message);

}

}

protected:

virtual void OnFormClosing(System::Windows::Forms::FormClosingEventArgs^ e) override

{

System::Windows::Forms::Form::OnFormClosing(e);

}

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

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

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

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::Label^ label5;

private: System::Windows::Forms::Label^ label6;

private: System::Windows::Forms::Label^ label7;

private: System::Windows::Forms::Label^ label8;

private: System::Windows::Forms::Label^ label9;

private: System::Windows::Forms::Label^ label10;

private: System::Windows::Forms::Label^ label11;

private: System::Windows::Forms::Label^ label12;

private: System::Windows::Forms::CheckBox^ checkBox1;

private: System::Windows::Forms::CheckBox^ checkBox2;

private: System::Windows::Forms::CheckBox^ checkBox3;

private: System::Windows::Forms::CheckBox^ checkBox4;

private: System::Windows::Forms::CheckBox^ checkBox5;

private: System::Windows::Forms::CheckBox^ checkBox6;

private: System::Windows::Forms::CheckBox^ checkBox7;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Label^ label13;

private: System::Windows::Forms::Label^ label14;

private: System::Windows::Forms::Label^ label15;

private: System::Windows::Forms::Label^ label16;

private: System::Windows::Forms::Label^ label17;

private: System::Windows::Forms::Label^ label18;

private: System::Windows::Forms::CheckBox^ checkBox8;

private: System::Windows::Forms::CheckBox^ checkBox9;

private: System::Windows::Forms::CheckBox^ checkBox10;

private: System::Windows::Forms::CheckBox^ checkBox11;

private: System::Windows::Forms::Label^ label19;

private: System::Windows::Forms::Label^ label20;

private: System::Windows::Forms::ComboBox^ comboBox1;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::Button^ button6;

protected:

protected:

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container^ components;

#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)

{

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

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

this->button1 = (gcnew System::Windows::Forms::Button());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->label5 = (gcnew System::Windows::Forms::Label());

this->label6 = (gcnew System::Windows::Forms::Label());

this->label7 = (gcnew System::Windows::Forms::Label());

this->label8 = (gcnew System::Windows::Forms::Label());

this->label9 = (gcnew System::Windows::Forms::Label());

this->label10 = (gcnew System::Windows::Forms::Label());

this->label11 = (gcnew System::Windows::Forms::Label());

this->label12 = (gcnew System::Windows::Forms::Label());

this->checkBox1 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox2 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox3 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox4 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox5 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox6 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox7 = (gcnew System::Windows::Forms::CheckBox());

this->button2 = (gcnew System::Windows::Forms::Button());

this->label13 = (gcnew System::Windows::Forms::Label());

this->label14 = (gcnew System::Windows::Forms::Label());

this->label15 = (gcnew System::Windows::Forms::Label());

this->label16 = (gcnew System::Windows::Forms::Label());

this->label17 = (gcnew System::Windows::Forms::Label());

this->label18 = (gcnew System::Windows::Forms::Label());

this->checkBox8 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox9 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox10 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox11 = (gcnew System::Windows::Forms::CheckBox());

this->label19 = (gcnew System::Windows::Forms::Label());

this->label20 = (gcnew System::Windows::Forms::Label());

this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->button6 = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 9);

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

this->label1->Size = System::Drawing::Size(132, 13);

this->label1->TabIndex = 0;

this->label1->Text = L"Параметры автозамены";

this->label1->Font = gcnew System::Drawing::Font(L"Arial", 8.0f, System::Drawing::FontStyle::Bold);

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(22, 48);

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

this->label2->Size = System::Drawing::Size(336, 13);

this->label2->TabIndex = 1;

this->label2->Text = L"Настройка замены и форматирования текста при вводе в Word:";

//

// button1

//

this->button1->Location = System::Drawing::Point(364, 43);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(150, 23);

this->button1->TabIndex = 2;

this->button1->Text = L"Параметры автозамены...";

this->button1->UseVisualStyleBackColor = true;

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(12, 22);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(373, 13);

this->label3->TabIndex = 3;

this->label3->Text = L"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

//

// label4

//

this->label4->AutoSize = true;

this->label4->Location = System::Drawing::Point(12, 81);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(313, 13);

this->label4->TabIndex = 4;

this->label4->Text = L"При исправлении орфографии в программах Microsoft Office";

this->label4->Font = gcnew System::Drawing::Font(L"Arial", 8.0f, System::Drawing::FontStyle::Bold);

//

// label5

//

this->label5->AutoSize = true;

this->label5->Location = System::Drawing::Point(12, 94);

this->label5->Name = L"label5";

this->label5->Size = System::Drawing::Size(373, 13);

this->label5->TabIndex = 5;

this->label5->Text = L"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

//

// label6

//

this->label6->AutoSize = true;

this->label6->Location = System::Drawing::Point(39, 116);

this->label6->Name = L"label6";

this->label6->Size = System::Drawing::Size(221, 13);

this->label6->TabIndex = 6;

this->label6->Text = L"Пропускать слова из ПРОПИСНЫХ БУКВ";

//

// label7

//

this->label7->AutoSize = true;

this->label7->Location = System::Drawing::Point(39, 136);

this->label7->Name = L"label7";

this->label7->Size = System::Drawing::Size(158, 13);

this->label7->TabIndex = 7;

this->label7->Text = L"Пропускать слова с цифрами";

//

// label8

//

this->label8->AutoSize = true;

this->label8->Location = System::Drawing::Point(39, 156);

this->label8->Name = L"label8";

this->label8->Size = System::Drawing::Size(257, 13);

this->label8->TabIndex = 8;

this->label8->Text = L"Пропускать адреса в Интернете и имена файлов";

//

// label9

//

this->label9->AutoSize = true;

this->label9->Location = System::Drawing::Point(39, 176);

this->label9->Name = L"label9";

this->label9->Size = System::Drawing::Size(173, 13);

this->label9->TabIndex = 9;

this->label9->Text = L"помечать повторяющеися слова";

//

// label10

//

this->label10->AutoSize = true;

this->label10->Location = System::Drawing::Point(39, 196);

this->label10->Name = L"label10";

this->label10->Size = System::Drawing::Size(346, 13);

this->label10->TabIndex = 10;

this->label10->Text = L"использовать прописыне с надстрочными знаками (французский)";

//

// label11

//

this->label11->AutoSize = true;

this->label11->Location = System::Drawing::Point(39, 216);

this->label11->Name = L"label11";

this->label11->Size = System::Drawing::Size(221, 13);

this->label11->TabIndex = 11;

this->label11->Text = L"Предлагать только из основного словоря";

//

// label12

//

this->label12->AutoSize = true;

this->label12->Location = System::Drawing::Point(39, 269);

this->label12->Name = L"label12";

this->label12->Size = System::Drawing::Size(239, 13);

this->label12->TabIndex = 12;

this->label12->Text = L"Русский: требовать точного использования ё";

//

// checkBox1

//

this->checkBox1->AutoSize = true;

this->checkBox1->Location = System::Drawing::Point(22, 116);

this->checkBox1->Name = L"1";

this->checkBox1->Size = System::Drawing::Size(15, 14);

this->checkBox1->TabIndex = 13;

this->checkBox1->UseVisualStyleBackColor = true;

int number = Int32::Parse(this->checkBox1->Name);

this->checkBox1->Checked = line[number - 1] == '1';

this->checkBox1->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox1\_CheckedChanged);

//

// checkBox2

//

this->checkBox2->AutoSize = true;

this->checkBox2->Location = System::Drawing::Point(22, 136);

this->checkBox2->Name = L"2";

this->checkBox2->Size = System::Drawing::Size(15, 14);

this->checkBox2->TabIndex = 14;

this->checkBox2->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox2->Name);

this->checkBox2->Checked = line[number - 1] == '1';

this->checkBox2->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox2\_CheckedChanged);

//

// checkBox3

//

this->checkBox3->AutoSize = true;

this->checkBox3->Location = System::Drawing::Point(22, 156);

this->checkBox3->Name = L"3";

this->checkBox3->Size = System::Drawing::Size(15, 14);

this->checkBox3->TabIndex = 15;

this->checkBox3->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox3->Name);

this->checkBox3->Checked = line[number - 1] == '1';

this->checkBox3->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox3\_CheckedChanged);

//

// checkBox4

//

this->checkBox4->AutoSize = true;

this->checkBox4->Location = System::Drawing::Point(22, 176);

this->checkBox4->Name = L"4";

this->checkBox4->Size = System::Drawing::Size(15, 14);

this->checkBox4->TabIndex = 16;

this->checkBox4->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox4->Name);

this->checkBox4->Checked = line[number - 1] == '1';

this->checkBox4->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox4\_CheckedChanged);

//

// checkBox5

//

this->checkBox5->AutoSize = true;

this->checkBox5->Location = System::Drawing::Point(22, 196);

this->checkBox5->Name = L"5";

this->checkBox5->Size = System::Drawing::Size(15, 14);

this->checkBox5->TabIndex = 17;

this->checkBox5->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox5->Name);

this->checkBox5->Checked = line[number - 1] == '1';

this->checkBox5->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox5\_CheckedChanged);

//

// checkBox6

//

this->checkBox6->AutoSize = true;

this->checkBox6->Location = System::Drawing::Point(22, 216);

this->checkBox6->Name = L"6";

this->checkBox6->Size = System::Drawing::Size(15, 14);

this->checkBox6->TabIndex = 18;

this->checkBox6->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox6->Name);

this->checkBox6->Checked = line[number - 1] == '1';

this->checkBox6->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox6\_CheckedChanged);

//

// checkBox7

//

this->checkBox7->AutoSize = true;

this->checkBox7->Location = System::Drawing::Point(22, 268);

this->checkBox7->Name = L"7";

this->checkBox7->Size = System::Drawing::Size(15, 14);

this->checkBox7->TabIndex = 19;

this->checkBox7->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox7->Name);

this->checkBox7->Checked = line[number - 1] == '1';

this->checkBox7->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox7\_CheckedChanged);

//

// button2

//

this->button2->Location = System::Drawing::Point(22, 239);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(153, 23);

this->button2->TabIndex = 20;

this->button2->Text = L"Настраиваемые словари...";

this->button2->UseVisualStyleBackColor = true;

//

// label13

//

this->label13->AutoSize = true;

this->label13->Location = System::Drawing::Point(12, 301);

this->label13->Name = L"label13";

this->label13->Size = System::Drawing::Size(209, 13);

this->label13->TabIndex = 21;

this->label13->Text = L"При исправлении правописания в Word";

this->label13->Font = gcnew System::Drawing::Font(L"Arial", 8.0f, System::Drawing::FontStyle::Bold);

//

// label14

//

this->label14->AutoSize = true;

this->label14->Location = System::Drawing::Point(12, 314);

this->label14->Name = L"label14";

this->label14->Size = System::Drawing::Size(373, 13);

this->label14->TabIndex = 22;

this->label14->Text = L"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

//

// label15

//

this->label15->AutoSize = true;

this->label15->Location = System::Drawing::Point(39, 338);

this->label15->Name = L"label15";

this->label15->Size = System::Drawing::Size(266, 13);

this->label15->TabIndex = 23;

this->label15->Text = L"Проверять орфографию в процессе набора текста";

//

// label16

//

this->label16->AutoSize = true;

this->label16->Location = System::Drawing::Point(39, 358);

this->label16->Name = L"label16";

this->label16->Size = System::Drawing::Size(321, 13);

this->label16->TabIndex = 24;

this->label16->Text = L"Отмечать грамматические ошибки в процессе набора текста";

//

// label17

//

this->label17->AutoSize = true;

this->label17->Location = System::Drawing::Point(39, 378);

this->label17->Name = L"label17";

this->label17->Size = System::Drawing::Size(87, 13);

this->label17->TabIndex = 25;

this->label17->Text = L"Сложные слова";

//

// label18

//

this->label18->AutoSize = true;

this->label18->Location = System::Drawing::Point(40, 398);

this->label18->Name = L"label18";

this->label18->Size = System::Drawing::Size(220, 13);

this->label18->TabIndex = 26;

this->label18->Text = L"Показывать статистику удобочитаемости";

//

// checkBox8

//

this->checkBox8->AutoSize = true;

this->checkBox8->Location = System::Drawing::Point(22, 338);

this->checkBox8->Name = L"8";

this->checkBox8->Size = System::Drawing::Size(15, 14);

this->checkBox8->TabIndex = 27;

this->checkBox8->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox8->Name);

this->checkBox8->Checked = line[number - 1] == '1';

this->checkBox8->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox8\_CheckedChanged);

//

// checkBox9

//

this->checkBox9->AutoSize = true;

this->checkBox9->Location = System::Drawing::Point(22, 358);

this->checkBox9->Name = L"9";

this->checkBox9->Size = System::Drawing::Size(15, 14);

this->checkBox9->TabIndex = 28;

this->checkBox9->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox9->Name);

this->checkBox9->Checked = line[number - 1] == '1';

this->checkBox9->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox9\_CheckedChanged);

//

// checkBox10

//

this->checkBox10->AutoSize = true;

this->checkBox10->Location = System::Drawing::Point(22, 378);

this->checkBox10->Name = L"10";

this->checkBox10->Size = System::Drawing::Size(15, 14);

this->checkBox10->TabIndex = 29;

this->checkBox10->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox10->Name);

this->checkBox10->Checked = line[number - 1] == '1';

this->checkBox10->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox10\_CheckedChanged);

//  
// checkBox11

//

this->checkBox11->AutoSize = true;

this->checkBox11->Location = System::Drawing::Point(22, 398);

this->checkBox11->Name = L"11";

this->checkBox11->Size = System::Drawing::Size(15, 14);

this->checkBox11->TabIndex = 30;

this->checkBox11->UseVisualStyleBackColor = true;

number = Int32::Parse(this->checkBox11->Name);

this->checkBox11->Checked = line[number - 1] == '1';

this->checkBox11->CheckedChanged += gcnew System::EventHandler(this, &MyForm::checkBox11\_CheckedChanged);

//

// label19

//

this->label19->AutoSize = true;

this->label19->Location = System::Drawing::Point(19, 418);

this->label19->Name = L"label19";

this->label19->Size = System::Drawing::Size(417, 13);

this->label19->TabIndex = 31;

this->label19->Text = L"Выбор проверки грамматики и стиля, которые будут выполняться Корректором";

//

// label20

//

this->label20->AutoSize = true;

this->label20->Location = System::Drawing::Point(19, 438);

this->label20->Name = L"label20";

this->label20->Size = System::Drawing::Size(81, 13);

this->label20->TabIndex = 32;

this->label20->Text = L"Набор правил:";

//

// comboBox1

//

this->comboBox1->FormattingEnabled = true;

this->comboBox1->Items->AddRange(gcnew cli::array< System::Object^ >(2) { L"Граматика и стиль", L"Граматика" });

this->comboBox1->Location = System::Drawing::Point(100, 435);

this->comboBox1->Name = L"12";

this->comboBox1->Size = System::Drawing::Size(130, 21);

this->comboBox1->TabIndex = 33;

number = Int32::Parse(this->comboBox1->Name);

if (line[number - 1] == '1') {

this->comboBox1->SelectedIndex = 1;

}

else {

this->comboBox1->SelectedIndex = 0;

}

comboBox1->SelectedIndexChanged += gcnew System::EventHandler(this, &MyForm::comboBox1\_SelectedIndexChanged);

//

// button3

//

this->button3->Location = System::Drawing::Point(236, 434);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(80, 23);

this->button3->TabIndex = 34;

this->button3->Text = L"Натсройка...";

this->button3->UseVisualStyleBackColor = true;

//

// button4

//

this->button4->Location = System::Drawing::Point(22, 462);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(125, 23);

this->button4->TabIndex = 35;

this->button4->Text = L"Повторная проверка";

this->button4->UseVisualStyleBackColor = true;

//

// button5

//

this->button5->Location = System::Drawing::Point(500, 560);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(85, 25);

this->button5->TabIndex = 35;

this->button5->Font = gcnew System::Drawing::Font(L"Arial", 10.0f, System::Drawing::FontStyle::Regular);

this->button5->Text = L"отмена";

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew EventHandler(this, &MyForm::button5\_Click);

//

// button6

//

this->button6->Location = System::Drawing::Point(400, 560);

this->button6->Name = L"button6";

this->button6->Size = System::Drawing::Size(85, 25);

this->button6->TabIndex = 35;

this->button6->Text = L"ок";

this->button6->Font = gcnew System::Drawing::Font(L"Arial", 12.0f, System::Drawing::FontStyle::Regular);

this->button6->UseVisualStyleBackColor = true;

this->button6->Click += gcnew EventHandler(this, &MyForm::button6\_Click);

//

// MyForm

//

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

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

this->ClientSize = System::Drawing::Size(600, 600);

this->Controls->Add(this->button6);

this->Controls->Add(this->button5);

this->Controls->Add(this->button4);

this->Controls->Add(this->button3);

this->Controls->Add(this->comboBox1);

this->Controls->Add(this->label20);

this->Controls->Add(this->label19);

this->Controls->Add(this->checkBox11);

this->Controls->Add(this->checkBox10);

this->Controls->Add(this->checkBox9);

this->Controls->Add(this->checkBox8);

this->Controls->Add(this->label18);

this->Controls->Add(this->label17);

this->Controls->Add(this->label16);

this->Controls->Add(this->label15);

this->Controls->Add(this->label14);

this->Controls->Add(this->label13);

this->Controls->Add(this->button2);

this->Controls->Add(this->checkBox7);

this->Controls->Add(this->checkBox6);

this->Controls->Add(this->checkBox5);

this->Controls->Add(this->checkBox4);

this->Controls->Add(this->checkBox3);

this->Controls->Add(this->checkBox2);

this->Controls->Add(this->checkBox1);

this->Controls->Add(this->label12);

this->Controls->Add(this->label11);

this->Controls->Add(this->label10);

this->Controls->Add(this->label9);

this->Controls->Add(this->label8);

this->Controls->Add(this->label7);

this->Controls->Add(this->label6);

this->Controls->Add(this->label5);

this->Controls->Add(this->label4);

this->Controls->Add(this->label3);

this->Controls->Add(this->button1);

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

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

this->Name = L"MyForm";

this->Text = L"MyForm";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void checkBox1\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void checkBox2\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void checkBox3\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void checkBox4\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void checkBox5\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void checkBox6\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}  
 private: System::Void checkBox7\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void checkBox8\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void checkBox9\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void checkBox10\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void checkBox11\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

CheckBox^ checkBox = dynamic\_cast<CheckBox^>(sender);

if (checkBox != nullptr) {

int number = Int32::Parse(checkBox->Name);

array<Char>^ chars = line->ToCharArray();

if (!checkBox->Checked) {

chars[number - 1] = '0';

}

else {

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

}

private: System::Void comboBox1\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

ComboBox^ comboBox = dynamic\_cast<ComboBox^>(sender);

int number = Int32::Parse(comboBox->Name);

array<Char>^ chars = line->ToCharArray();

if (comboBox->SelectedIndex == 0) {

chars[number - 1] = '0';

}

else if(comboBox->SelectedIndex == 1){

chars[number - 1] = '1';

}

line = gcnew String(chars);

}

private: System::Void button6\_Click(Object^ sender, EventArgs^ e)

{

SaveFile();

this->Close();

}

private: System::Void button5\_Click(Object^ sender, EventArgs^ e)

{

this->Close();

}

};

}

Итог:

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

Контент, сгенерированный ИИ, может содержать ошибки.

Реализовал форму, также добавил способность запоминать настройки.

**Вывод:** в ходе проделанной работы я научился пользоваться средой разработки Visual Studio, создавать проекты, также научился создавать формы с помощью Windows Forms.