Выполнил

студент группы КТбо1-2 C. С. Абрамов

Принял

доцент кафедры САиТ В. С. Лапшин

Таганрог 2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

(ФГАОУ ВО «ЮФУ»)

Институт компьютерных технологий и информационной безопасности

Кафедра системного анализа и телекоммуникаций

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине «Основы алгоритмизации и программирования»

**«РАЗРАБОТКА ПРИЛОЖЕНИЙ НА С++   
С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ»**

Вариант 1

# Содержание

Содержание 2

Техническое задание 3

Цель задания 3

Задача 3

Ход работы 4

Написание программы 4

Пример работы программы 5

Вывод 6

Листинг 7

# Техническое задание

## Цель задания

Цель задания: разработать приложение с графическим интерфейсом для Windows, используя библиотеку .NET (CLI-приложение).

## Задача

Задачи лабораторной работы:

1. Создать проект WinForms в среде Visual Studio;
2. Включить класс, разработанный в лабораторной работе №4, в созданный проект;
3. Разработать форму ввода данных, контроля введенной информации и вывода информации на экран. Функции работы с информацией взять из подключенного класса;
4. Отладить приложение.

# Ход работы

## Написание программы

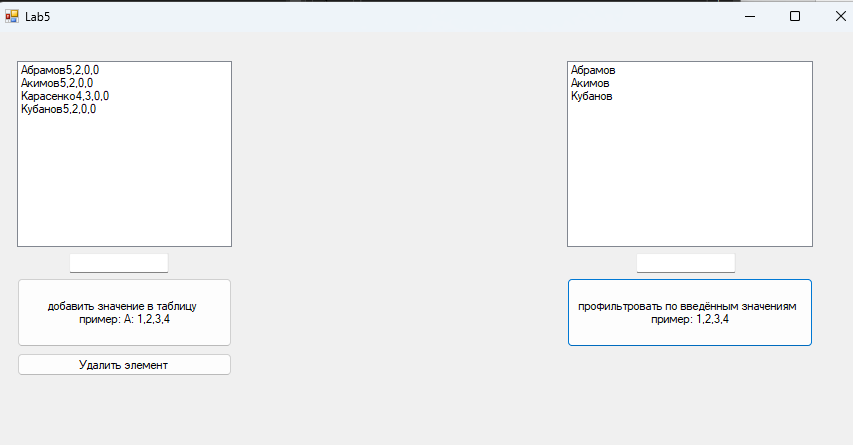
Для создания WinForms приложения я воспользовался шаблоном CLR. Для этого я создал пустое приложение через «Решение проекта» - «Создать элемент» - «Среда CLR» - «Форма WinForms».

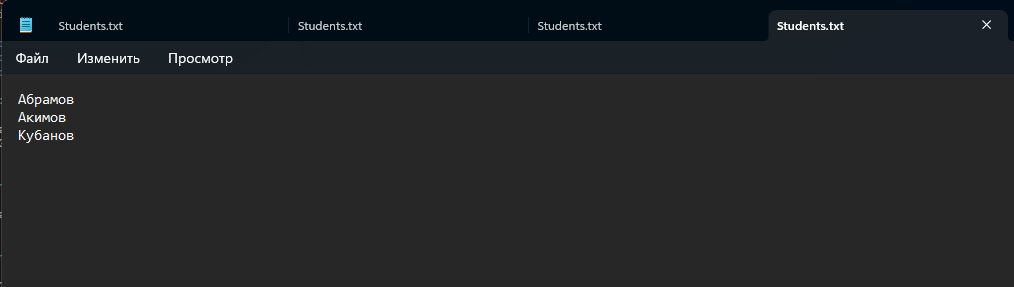
Далее через конструктор VS я добавил приложению кнопки Button, список ListView и текстовые поля TextBox. Visual Studio автоматически собрал необходимый код в MyForm.h, с которым можно работать далее.

Логика программы следующая:

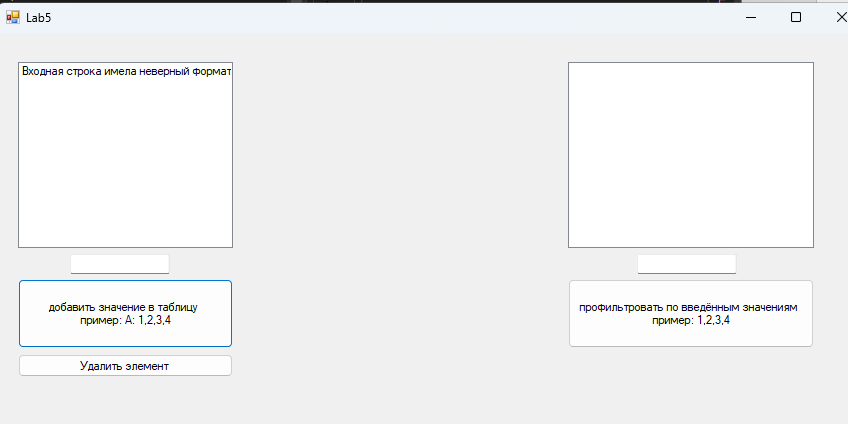
1. После запуска приложения пользователь вписывает данные, с которыми он будет в дальнейшем работать. Далее пользователь может делать следующее:
2. Пользователь может найти ввести информацию. Для этого, в первое текстовое поле над кнопкой “добавить значение в таблицу \n пример: А: 1,2,3,4” выбора категорию он вводит данные. После нажатия кнопки программа вписывает значение в ListBox1 и добавляет её в приватный переменный класс.
   1. . Если какой-то из файлов не может быть прочтен, пользователю в listBox1 будет выдана ошибка.
3. Пользователь может удалить элемент из переменной класса и listBox1 через кнопку “Удалить элемент”.
4. Пользователь может отфильтровать введённую информацию, введя искомое значение под listBox2, после чего в listBox2 будет выведен ответ программы.
5. Файлы, отфильтрованные в listBox2 добавляются в файл Students.txt.

## Пример работы программы





“*При удачной работе*”



*“В случае ошибки при вводе”*

# Вывод

Итогом работы стало приложение WinForms с графическим интерфейсом, позволяющее управлять данными о оценках студентов. Пользователь может найти данные, добавить/удалить их.

В ходе работы я воспользовался реализованным классом FileHandler из лабораторной работы №4, а также ознакомился с возможностями конструктора CLI приложения Visual Studio, и реализовал свое приложение с графическим интерфейсом.

# Листинг

## Lab5.h

#pragma once

#include "FileHandler.h"

#include "pch.h"

#include "iostream"

//#include <cliext/utility>

namespace ALab5 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для Lab5

/// </summary>

public ref class Lab5 : public System::Windows::Forms::Form

{

public:

Lab5(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Lab5()

{

if (components)

{

delete components;

}

}

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

private: System::Windows::Forms::ListBox^ listBox1;

private: System::Windows::Forms::TextBox^ textBox2;

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

private: System::Windows::Forms::ListBox^ listBox2;

private: System::Windows::Forms::TextBox^ textBox1;

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

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

this->listBox1 = (gcnew System::Windows::Forms::ListBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

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

this->listBox2 = (gcnew System::Windows::Forms::ListBox());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

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

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(21, 246);

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

this->button1->Size = System::Drawing::Size(215, 69);

this->button1->TabIndex = 0;

this->button1->Text = L"добавить значение в таблицу \n пример: А: 1,2,3,4";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &Lab5::button1\_Click);

//

// listBox1

//

this->listBox1->FormattingEnabled = true;

this->listBox1->Location = System::Drawing::Point(21, 29);

this->listBox1->Name = L"listBox1";

this->listBox1->Size = System::Drawing::Size(215, 186);

this->listBox1->TabIndex = 2;

this->listBox1->SelectedIndexChanged += gcnew System::EventHandler(this, &Lab5::listBox1\_SelectedIndexChanged);

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(73, 221);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(100, 20);

this->textBox2->TabIndex = 3;

this->textBox2->TextChanged += gcnew System::EventHandler(this, &Lab5::textBox2\_TextChanged);

//

// button2

//

this->button2->Location = System::Drawing::Point(571, 246);

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

this->button2->Size = System::Drawing::Size(246, 69);

this->button2->TabIndex = 4;

this->button2->Text = L"профильтровать по введённым значениям \n пример: 1,2,3,4";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &Lab5::button2\_Click);

//

// listBox2

//

this->listBox2->FormattingEnabled = true;

this->listBox2->Location = System::Drawing::Point(571, 29);

this->listBox2->Name = L"listBox2";

this->listBox2->Size = System::Drawing::Size(246, 186);

this->listBox2->TabIndex = 5;

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(640, 221);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(100, 20);

this->textBox1->TabIndex = 6;

//

// button3

//

this->button3->Location = System::Drawing::Point(21, 321);

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

this->button3->Size = System::Drawing::Size(215, 23);

this->button3->TabIndex = 7;

this->button3->Text = L"Удалить элемент";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &Lab5::button3\_Click);

//

// Lab5

//

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

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

this->ClientSize = System::Drawing::Size(868, 508);

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

this->Controls->Add(this->textBox1);

this->Controls->Add(this->listBox2);

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

this->Controls->Add(this->textBox2);

this->Controls->Add(this->listBox1);

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

this->Name = L"Lab5";

this->Text = L"Lab5";

this->Load += gcnew System::EventHandler(this, &Lab5::Lab5\_Load);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Windows::Forms::Timer^ timer;

FileHandler^ fh = gcnew FileHandler();

List<int>^ ParseGrades(String^ gradesStr) {

List<int>^ grades = gcnew List<int>();

array<String^>^ gradesArray = gradesStr->Split(',');

for each (String ^ grade in gradesArray) {

grades->Add(Int32::Parse(grade->Trim()));

}

return grades;

}

private: System::Void Lab5\_Load(System::Object^ sender, System::EventArgs^ e) {

}

String^ name;

System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

String^ inputText = this->textBox2->Text;

if (!String::IsNullOrEmpty(inputText)) {

/\*this->listBox1->Items->Add(inputText);\*/

array<String^>^ parts = inputText->Split(':');

if (parts->Length == 2) {

name = parts[0]->Trim();

try {

List<int>^ grades = ParseGrades(parts[1]->Trim());

fh->AddStudent(name, grades);

this->listBox1->Items->Add(name + parts[1]->Trim());

}

catch(Exception^ ex){

this->listBox1->Items->Add(ex->Message);

}

}

this->textBox2->Clear();

}

}

System::Void listBox1\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

}

System::Void textBox2\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

}

/\*private: System::Boolean IsAlphabeticString(String^ input) {

System::Text::RegularExpressions::Regex^ regex = gcnew System::Text::RegularExpressions::Regex("^[a-zA-Z]+$");

return regex->IsMatch(input);

}\*/

private: System::Boolean IsNumericCommaSeparatedString(String^ input) {

System::Text::RegularExpressions::Regex^ regex = gcnew System::Text::RegularExpressions::Regex("^[0-9]+(,[0-9]+)\*$");

return regex->IsMatch(input);

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

this->listBox2->Items->Clear();

String^ inputText = this->textBox1->Text;

if ((!String::IsNullOrEmpty(inputText)) && (IsNumericCommaSeparatedString(inputText))) {

Dictionary<String^, List<int>^>^ studentsGrades = fh->GetName();

int countFives = 0;

int countFours = 0;

int countThrees = 0;

int countTwos = 0;

List<int>^ inputGrades = ParseGrades(inputText);

for each (int grade in inputGrades)

{

if (grade == 5) countFives++;

else if (grade == 4) countFours++;

else if (grade == 3) countThrees++;

else if (grade == 2) countTwos++;

}

ArrayList^ Mstudents = fh->GetStudentsByGrades(countFives, countFours, countThrees, countTwos);

System::IO::StreamWriter^ sw = gcnew System::IO::StreamWriter(this->fh->TextName());

for each (String ^ student in Mstudents)

{

sw->WriteLine(student);

this->listBox2->Items->Add(student);

}

sw->Close();

this->textBox1->Clear();

}

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

if (listBox1->Items->Count > 0) {

this->listBox1->Items->RemoveAt(listBox1->Items->Count - 1);

fh->DelStudent(name);

}

}

};

}

## Filehandler.cpp

#include "pch.h"

#include "FileHandler.h"

using namespace System;

using namespace System::Collections;

using namespace System::Collections::Generic;

FileHandler::FileHandler(void)

{

TextFileName = gcnew String("Students.txt");

studentGrades = gcnew Dictionary<String^, List<int>^>();

}

void FileHandler::SetTextFileName(String^ NewName)

{

TextFileName = NewName;

}

void FileHandler::AddStudent(String^ name, List<int>^ grades)

{

studentGrades[name] = grades;

}

void FileHandler::DelStudent(String^ name)

{

studentGrades->Remove(name);

}

Dictionary<String^, List<int>^>^ FileHandler::GetName()

{

return studentGrades;

}

String^ FileHandler::TextName() {

return TextFileName;

}

ArrayList^ FileHandler::GetStudentsByGrades(int fives, int fours, int threes, int twos)

{

ArrayList^ result = gcnew ArrayList();

for each (KeyValuePair<String^, List<int>^> student in studentGrades)

{

int countFives = 0;

int countFours = 0;

int countThrees = 0;

int countTwos = 0;

for each (int grade in student.Value)

{

if (grade == 5) countFives++;

else if (grade == 4) countFours++;

else if (grade == 3) countThrees++;

else if (grade == 2) countTwos++;

}

if (countFives == fives && countFours == fours && countThrees == threes && countTwos == twos)

{

result->Add(student.Key);

}

}

return result;

}

void FileHandler::SaveToFileStudentsByGrades(int fives, int fours, int threes, int twos)

{

ArrayList^ students = GetStudentsByGrades(fives, fours, threes, twos);

for each (String ^ student in students)

{

System::Console::WriteLine(student);

}

System::IO::StreamWriter^ sw = gcnew System::IO::StreamWriter(this->TextFileName);

for each (String ^ student in students)

{

sw->WriteLine(student);

}

sw->Close();

}

## lab5.cpp

#include "Lab5.h"

using namespace System;

using namespace System::Windows::Forms;

using namespace ALab5;

[STAThread]

int main(array<String^>^ args) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Application::Run(gcnew Lab5());

return 0;

}

## filehandler.h

using namespace System;

using namespace System::Collections;

using namespace System::Collections::Generic;

ref class FileHandler

{

public:

// Конструктор класса

FileHandler(void);

// Добавить студента с оценками в список

void AddStudent(String^, List<int>^);

// Удалить студента из списка

void DelStudent(String^);

// Установить имя файла для сохранения результатов поиска

void SetTextFileName(String^);

// Получить список студентов с заданным количеством оценок

ArrayList^ GetStudentsByGrades(int, int, int, int);

// Сохранить список студентов с заданным количеством оценок

void SaveToFileStudentsByGrades(int, int, int, int);

Dictionary<String^, List<int>^>^ FileHandler::GetName();

String^ FileHandler::TextName();

private:

String^ TextFileName;

Dictionary<String^, List<int>^>^ studentGrades;

};

## lab5.h[конструкт]