Выполнили:

студенты группы 19ВВ2

Гусев Д.О.

Кубасво И.М.

Приняли:

Синев М.П.

Дорошенко И.Н.

Пенза 2022

ОТЧЕТ

по лабораторной работе №6

по курсу «Технологии программирования»

на тему «Стандартные коллекции и файловые потоки»

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Цель работы

Изучение механизма вызова функций из внешних библиотек. Получение навыков передачи параметров и возврата результата из функций внешних библиотек.

### Лабораторное задание

Лабораторное задание выполняется на базе лабораторной работы №1.

В работе требуется реализовать программу, состоящую из DLL библиотеки и консольного приложения. Программа должна выполнять задание, указанное в лабораторной работе №1.

Консольное приложение должно реализовать набор классов, приведенный в лабораторной работе №1. Во всех функция, выполняющих ввод/вывод данных, вместо стандартных функций должны использоваться вызовы из DLL библиотеки.

Листинг:

Файл DLL1.dll

// pch.cpp: файл исходного кода, соответствующий предварительно скомпилированному заголовочному файлу

#include "pch.h"

// При использовании предварительно скомпилированных заголовочных файлов необходим следующий файл исходного кода для выполнения сборки.

#include <string>

#include <iostream>

#include <Windows.h>

#include <locale>

#include <codecvt>

#include <tchar.h>

class Book

{

public:

wchar\_t Theme[100]; // Тема

wchar\_t Author[100]; // Автор

wchar\_t Year[10]; // Год издания

};

typedef struct Derived11

{

wchar\_t Town[100]; // Город

}Derived1;

typedef struct Derived22

{

wchar\_t Genre[100]; // Жанр

}Derived2;

typedef struct Derived33

{

float Rating = 0.0; // Рейтинг

}Derived3;

// Base

// Theme

extern "C" void \_\_declspec(dllexport) \_cdecl InputTheme(Book \* b1)

{

printf(" Theme: ");

wscanf\_s(L"%s", b1->Theme, (unsigned)\_countof(b1->Theme));

}

extern "C" void \_\_declspec(dllexport) \_cdecl OutputTheme(Book \* b1)

{

\_tprintf(L" Theme: %s \n", b1->Theme);

}

// Author

extern "C" void \_\_declspec(dllexport) \_cdecl InputAuthor(Book \* b1)

{

printf(" Author: ");

wscanf\_s(L"%s", b1->Author, (unsigned)\_countof(b1->Author));

}

extern "C" void \_\_declspec(dllexport) \_cdecl OutputAuthor(Book \* b1)

{

\_tprintf(L" Author: %s \n", b1->Author);

}

// Year

extern "C" void \_\_declspec(dllexport) \_cdecl InputYear(Book \* b1)

{

printf(" Year: ");

wscanf\_s(L"%s", &(b1->Year), (unsigned)\_countof(b1->Year));

}

extern "C" void \_\_declspec(dllexport) \_cdecl OutputYear(Book \* b1)

{

\_tprintf(L" Year: %s \n", b1->Year);

}

//Derived1

//Town

extern "C" void \_\_declspec(dllexport) \_cdecl InputDerived1(Derived1 \* d1)

{

printf(" Town: ");

wscanf\_s(L"%s", &(d1->Town), (unsigned)\_countof(d1->Town));

}

extern "C" void \_\_declspec(dllexport) \_cdecl OutputDerived1(Derived1 \* d1)

{

\_tprintf(L" Town: %s\n", d1->Town);

}

//Derived2

//Genre

extern "C" void \_\_declspec(dllexport) \_cdecl InputDerived2(Derived2 \* d2)

{

printf("Genre: ");

wscanf\_s(L"%s", &(d2->Genre), (unsigned)\_countof(d2->Genre));

}

extern "C" void \_\_declspec(dllexport) \_cdecl OutputDerived2(Derived2 \* d2)

{

\_tprintf(L"Genre: %s\n", d2->Genre);

}

//Derived3

//Rating

extern "C" void \_\_declspec(dllexport) \_cdecl InputDerived3(Derived3 \* d3)

{

printf("Rating: ");

wscanf\_s(L"%f", &(d3->Rating));

}

extern "C" void \_\_declspec(dllexport) \_cdecl OutputDerived3(Derived3 \* d3)

{

\_tprintf(L"Rating: %f\n", d3->Rating);

}

Файл Program.ch

using System;

using System.Runtime.InteropServices;

using System.Text;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

//4. Класс «Книга» (тема, автор, издательство, год выпуска и т.д.).

namespace lab6

{

class Program

{

// Derived1

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void InputDerived1(ref Derived1 d1);

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void OutputDerived1(ref Derived1 d1);

// Derived2

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void InputDerived2(ref Derived2 d2);

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void OutputDerived2(ref Derived2 d2);

// Derived3

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void InputDerived3(ref Derived3 d3);

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void OutputDerived3(ref Derived3 d3);

// Book

//Theme

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void InputTheme([In, Out] Book b1);

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void OutputTheme([In, Out] Book b1);

//Author

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void InputAuthor([In, Out] Book b1);

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void OutputAuthor([In, Out] Book b1);

//Year

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void InputYear([In, Out] Book b1);

[DllImport("DLL1.dll", CallingConvention = CallingConvention.Cdecl,

CharSet = CharSet.Unicode)]

public static extern void OutputYear([In, Out] Book b1);

static int Main(string[] args)

{

Derived1 d1 = new();

Derived2 d2 = new();

Derived3 d3 = new();

Book b1 = new();

// Book

Console.WriteLine("Input Book:");

InputTheme(b1);

InputAuthor(b1);

InputYear(b1);

Console.WriteLine("Output Book:");

OutputTheme(b1);

OutputAuthor(b1);

OutputYear(b1);

// Derived1

Console.WriteLine("\n\nInput Derived1:");

InputDerived1(ref d1);

Console.WriteLine("Output Derived1:");

OutputDerived1(ref d1);

// Derived2

Console.WriteLine("\n\nInput Derived2:");

InputDerived2(ref d2);

Console.WriteLine("Output Derived2:");

OutputDerived2(ref d2);

// Derived3

Console.WriteLine("\n\nInput Derived3:");

InputDerived3(ref d3);

Console.WriteLine("Output Derived3:");

OutputDerived3(ref d3);

return 0;

}

}

}

Файл Book.ch

using System;

using System.Runtime.InteropServices;

namespace lab6

{

[StructLayout(LayoutKind.Sequential, CharSet = CharSet.Unicode)]

public class Book

{

[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 100)]

public string Theme;

[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 100)]

public string Author;

[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 100)]

public string Year;

}

}

Файл Derived1.ch

using System.Runtime.InteropServices;

namespace lab6

{

[StructLayout(LayoutKind.Sequential, CharSet = CharSet.Unicode)]

public struct Derived1

{

[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 100)]

public string Town;

}

}

Файл Derived2.ch

using System.Runtime.InteropServices;

namespace lab6

{

[StructLayout(LayoutKind.Sequential, CharSet = CharSet.Unicode)]

public struct Derived2

{

[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 100)]

public string Genre; // Адрес магазина

}

}

Файл Derived3.ch

using System.Runtime.InteropServices;

namespace lab6

{

[StructLayout(LayoutKind.Sequential, CharSet = CharSet.Unicode)]

public struct Derived3

{

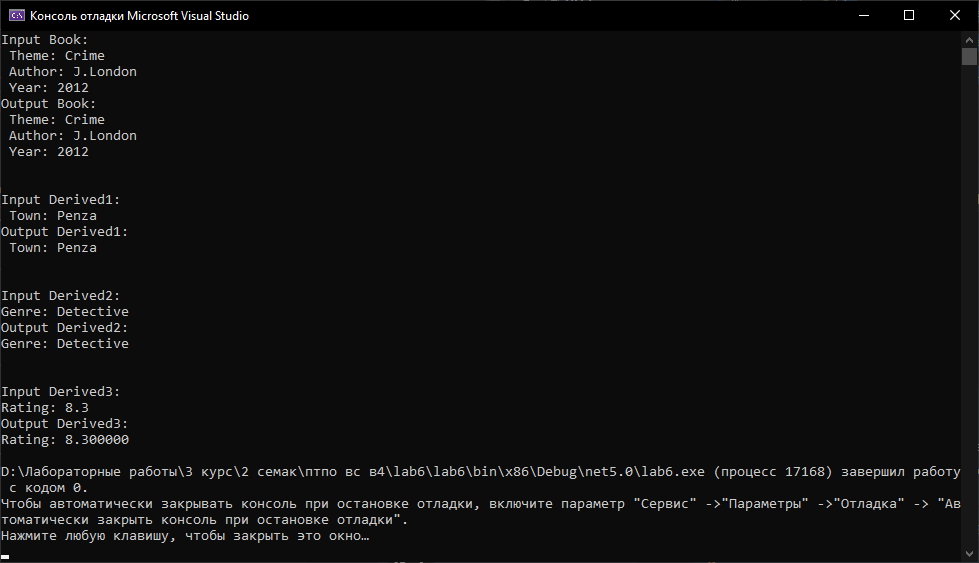
[MarshalAs(UnmanagedType.R4, SizeConst = 100)]

public float Rating;

}

}

### Результат работы



**Вывод:** Мы изучили механизм вызова функций из внешних библиотек. Получили навыки передачи параметров и возврата результата из функций внешних библиотек.