

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА**  
**ФРАНКА**

Факультет прикладної математики та інформатики

Кафедра програмування

**Лабораторна робота № 6**

Використання DLL в C++

Виконала  
студентка групи ПМО-41  
Кравець Ольга

2024

## Хід роботи

Взяла за основу бібліотеку DLL, створену в лабораторній роботі 5: MyLib.dll та MyLib.lib. Ці файли скопіювала в свій каталог, до якого буде доступ з різних проєктів.

... Практичні > DLL lab 5,6,7 > 06 > MyLib				
Сортування ▾ Подання ▾ ...				
Ім'я	Дата змінення	Тип	Розмір	
MyLib.dll	15.03.2024 22:16	Розширення заст...	59 КБ	
MyLib.lib	15.03.2024 22:12	Object File Library	3 КБ	

Склала тестову програму за проєктом “Empty Project”, яка викликає функції DLL. Побудувала виклики методом **неявного зв’язування**.

Неявне зв’язування з DLL виконують на етапі компіляції і будови прикладної програми користувача. Середовище програмування самостійно виконує пошук бібліотеки і зв’язує потрібні функції DLL з прикладною програмою. В цьому разі були необхідні дві речі: назва файла імпорту і шлях до розташування файла імпорту.

Отже, налаштувала зв’язок з файлом імпорту \*.lib; налаштувала зв’язок з файлом бібліотеки \*.dll. Виконала налаштування за інструкцією:

Створила Empty Project, куди імпортувала прототипи функцій з DLL, а також перенесла функцію main() з тестами з лабораторної роботи 5.

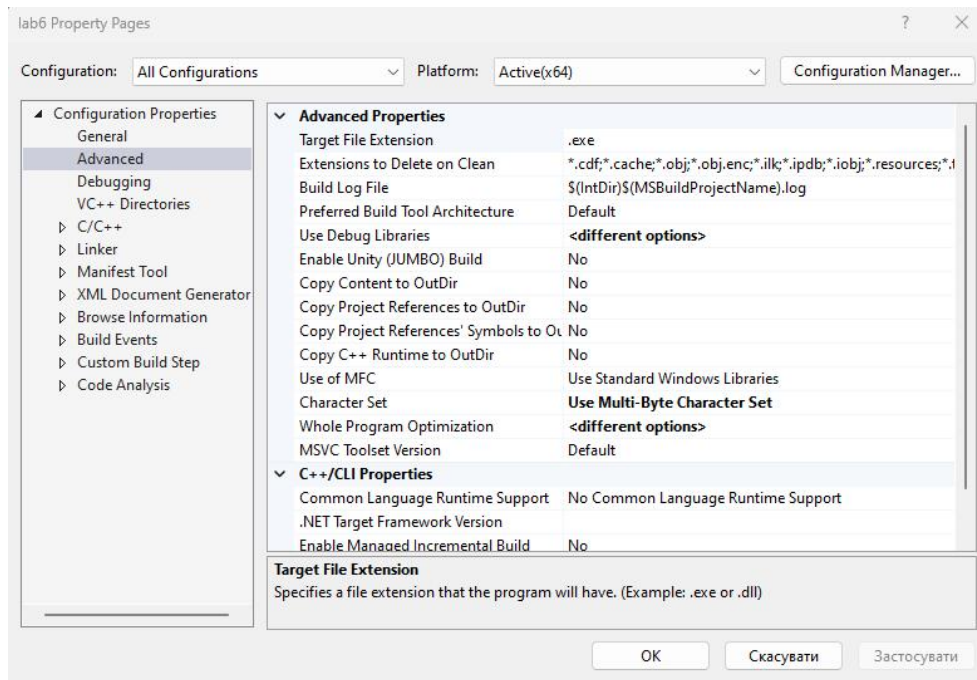
```
Source.cpp -# X
lab6 (Global Scope)
1 #include "mytypesbase.h"
2 #include <assert.h>
3 #include <iostream>
4 using namespace std;
5
6 extern "C" __declspec(dllexport) INT(__stdcall FindMinimum) (LPINT s, INT arrSize);
7 extern "C" __declspec(dllexport) INT(__stdcall CalculateFactorial) (INT n);
8 extern "C" __declspec(dllexport) BOOL ContainsSub(LPSTR toCheck, INT checkedSize, LPSTR toFind, INT foundSize, BOOL caseSensitive);
9 extern "C" __declspec(dllexport) DOUBLE AverageValue(LPINT numArr, INT arrSize);
10
11 int main()
12 {
13     // Тестування функції FindMinimum
14     int arrFindMin[5] = { 5, -3, 8, 1, 0 };
15     int minResult = FindMinimum(arrFindMin, sizeof(arrFindMin));
16     assert(minResult == -3);
17     cout << "FindMinimum function test passed." << endl;
18
19     // Тестування функції CalculateFactorial
20     // Тест 1: Позитивне число
21     int n1 = 5;
22     int expected1 = 120;
23     assert(CalculateFactorial(n1) == expected1);
24
25     // Тест 2: Нуль
26     int n2 = 0;
27     int expected2 = 1;
28     assert(CalculateFactorial(n2) == expected2);
29
30     // Тест 3: Від'ємне число
31     int n3 = -3;
32     int expected3 = -1;
33     assert(CalculateFactorial(n3) == expected3);
34     cout << "CalculateFactorial tests passed." << endl;
35
36     // Тестування функції ContainsSub
37     char toCheck[] = "Hello, World!";
38     char toFind[] = "World";
39     bool containsResult = ContainsSub(toCheck, sizeof(toCheck), toFind, sizeof(toFind), true);
40     assert(containsResult);
41
42     char toCheck2[] = "Hello, World!";
43     char toFind2[] = "world"; // Маленька 'w', оскільки регістр не враховується
44     bool containsResult2 = ContainsSub(toCheck, sizeof(toCheck), toFind, sizeof(toFind), false);
45     assert(containsResult2);
46     cout << "ContainsSub function tests passed." << endl;
47
48     // Тестування функції AverageValue
49     int numArr[5] = { 3, 5, 7, 10, 2 }; // Average = 5.4
50     double averageResult = AverageValue(numArr, sizeof(numArr));
51     assert(averageResult == 5.4);
52     cout << "AverageValue function test passed." << endl;
53
54     return 0;
55 }
```

На цьому кроці помилок у самому коді не було, але після запуску компілятор показав декілька помилок зв'язані з поганими зовнішніми посиланнями на функції.

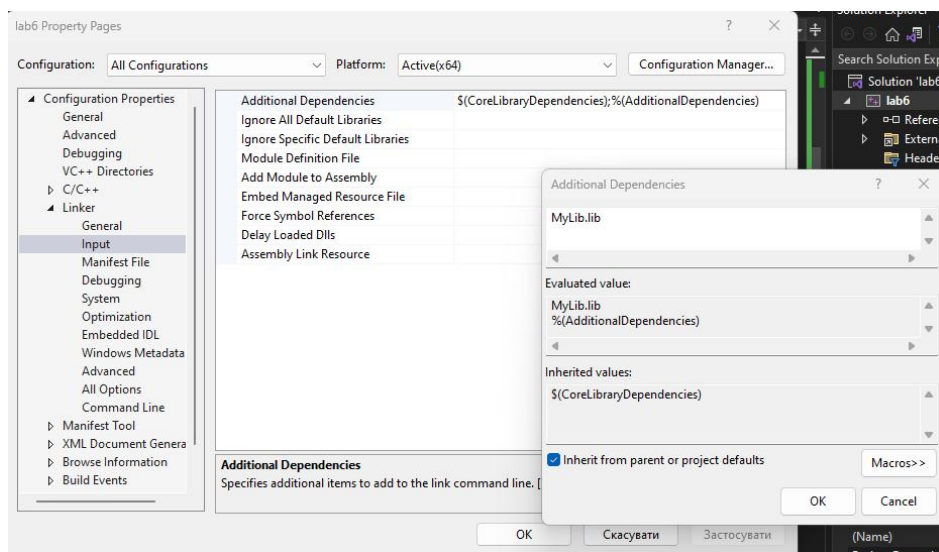
```
Error List
Entire Solution 5 Errors 0 Warnings 0 Messages Build + IntelliSense Search Error List
Code Description Project File Supp...
LNK2019 unresolved external symbol FindMinimum referenced in function main lab6 1 Source.obj
LNK2019 unresolved external symbol CalculateFactorial referenced in function main lab6 1 Source.obj
LNK2019 unresolved external symbol ContainsSub referenced in function main lab6 1 Source.obj
LNK2019 unresolved external symbol AverageValue referenced in function main lab6 1 Source.obj
LNK1120 4 unresolved externals lab6 1 lab6.exe
```

Причиною цих помилок було те, що компонувальнику потрібний файл MyLib.lib. Отже, для виконання налаштування для пошуку цього файлу були виконані такі дії:

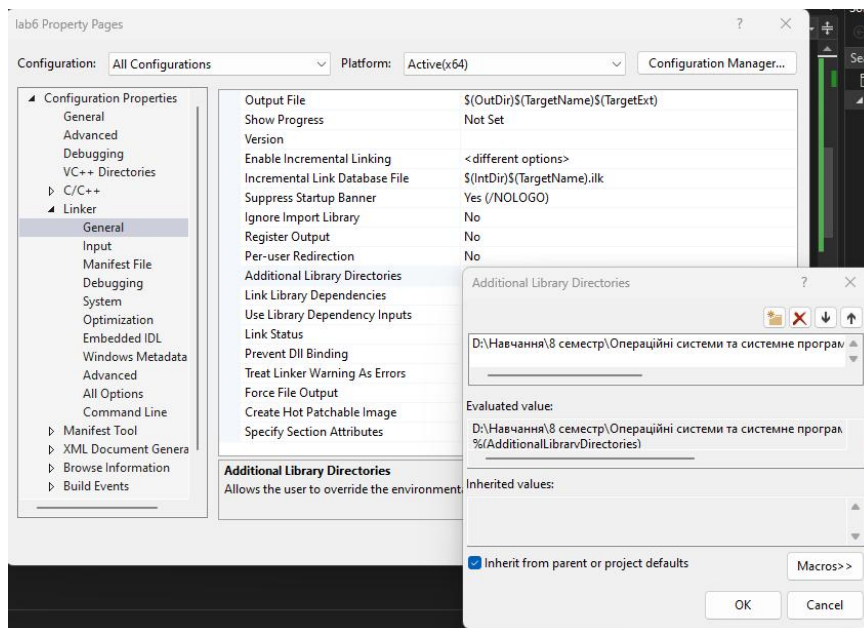
- Зайшла у вікно Property Pages через Visual Studio (Project -> Properties).



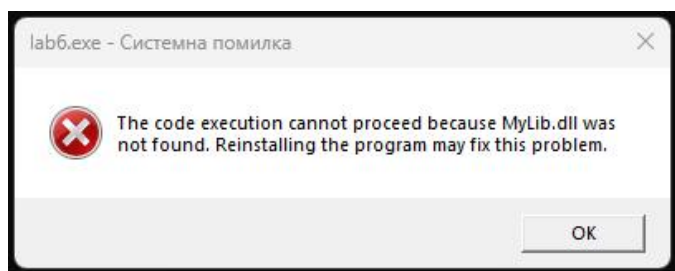
- На лівій панелі вибрала Configuration Properties -> Linker-> Input та на панелі властивостей обрала Additional Dependencies, де і додала MyLib.lib файл.



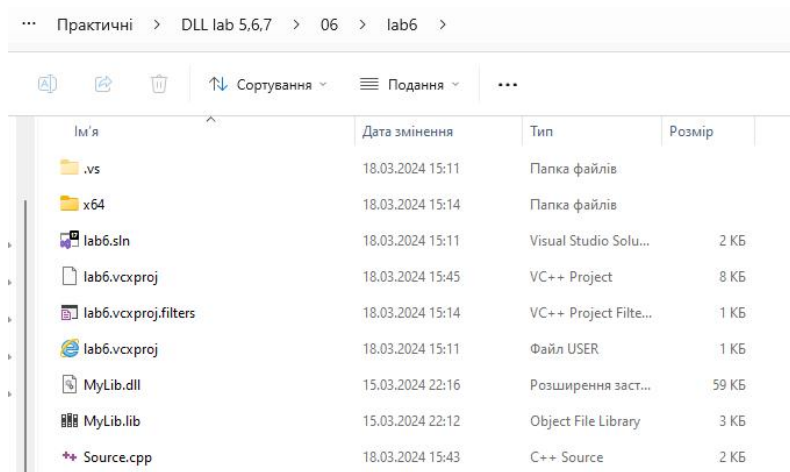
- Після цього зайшла у вкладку Configuration Properties -> Linker -> General та на панелі властивостей обрала Additional Dependencies, де записала шлях до розташування файлу імпорту (У моєму випадку файл був перенесений у папку D:\Навчання\8 семестр\Операційні системи та системне програмування\Практичні\DLL lab 5,6,7\06\MyLib)



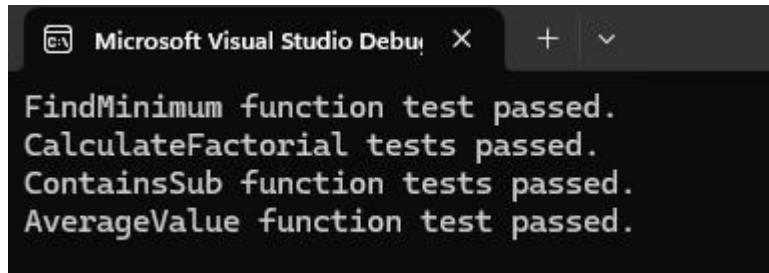
Тепер можна запуснути програму без помилок від компілятора, проте після запуску програма “вилітає” з такою ПОМИЛКОЮ:



Це свідчить про те, що під час виконання операційна система намагалася знайти саме бібліотеку MyLib.dll, але не змогла. Найпростіший спосіб уникнути цю проблему – помістити цей файл у папку, де знаходиться exe файл проєкту, що я і зробила.



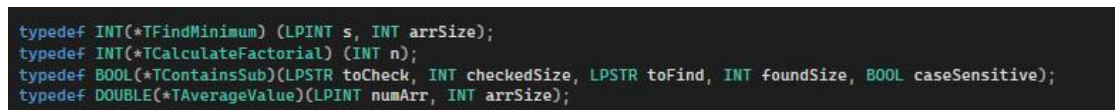
Після цього програма успішно запуситься в дебагері. Для тестування використала ті самі методи, що і в лабораторній роботі 5(вище код наведено) і всі пройшли успішно.



```
Microsoft Visual Studio Debug Console
FindMinimum function test passed.
CalculateFactorial tests passed.
ContainsSub function tests passed.
AverageValue function test passed.
```

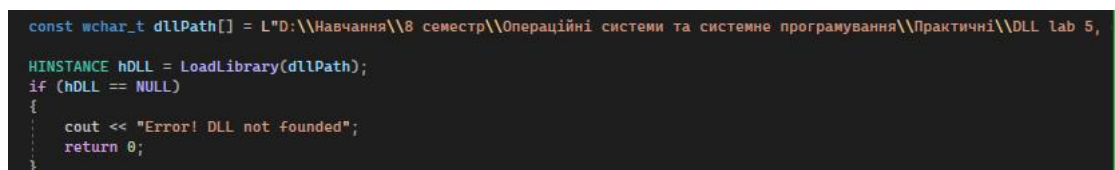
Склала другий проєкт “Empty Project”, який би мав виклики функцій DLL методом **явного зв’язування**. Використала ту саму тестову програму, що й для неявного зв’язування.

Для цього створила ще один Empty Project і перенесла туди прототипи функцій з додаванням ключового слова `typedef`. Завдяки цьому можна оголосити спеціальні типи вказівників на зовнішні функції DLL.



```
typedef INT(*TFindMinimum)(LPINT s, INT arrSize);
typedef INT(*TCalculateFactorial)(INT n);
typedef BOOL(*TContainsSub)(LPSTR toCheck, INT checkedSize, LPSTR toFind, INT foundSize, BOOL caseSensitive);
typedef DOUBLE(*TAverageValue)(LPINT numArr, INT arrSize);
```

Після цього вказала в програму шлях до файлу MyLib.dll і дала цей шлях як параметр в функцію `LoadLibrary`, яка зможе імпортувати бібліотеку у цю програму.



```
const wchar_t dllPath[] = L"D:\\Навчання\\8 семестр\\Операційні системи та системне програмування\\Практичні\\DLL lab 5,
HINSTANCE hDLL = LoadLibrary(dllPath);
if (hDLL == NULL)
{
    cout << "Error! DLL not founded";
    return 0;
}
```

Після цього побудувала вказівник на певну функцію завдяки `GetProcAddress`, яка може імпортувати потрібні нам функції з DLL. Також додала відповідні перевірки як для бібліотеки, так і для окремих функцій і видалила створені вказівники за



допомогою функції FreeLibrary. Тоді перенесла тести з минулої лабораторної. У результаті отримала такий код:

```
11 int main()
12 {
13     const wchar_t dllPath[] = L"C:\\MyLib\\MyLib.dll";
14     HINSTANCE hDLL = LoadLibrary(dllPath);
15     if (hDLL == NULL)
16     {
17         cout << "Error! DLL not founded";
18         return 0;
19     }
20
21     TFindMinimum findMinimum = (TFindMinimum)GetProcAddress(hDLL, "FindMinimum");
22     if (!findMinimum)
23     {
24         cout << "Function FindMinimum not founded";
25         return 0;
26     }
27
28     TCalculateFactorial calculateFactorial = (TCalculateFactorial)GetProcAddress(hDLL, "CalculateFactorial");
29     if (!calculateFactorial)
30     {
31         cout << "Function CalculateFactorial not founded";
32         return 0;
33     }
34
35     TContainsSub containsSub = (TContainsSub)GetProcAddress(hDLL, "ContainsSub");
36     if (!containsSub)
37     {
38         cout << "Function ContainsSub not founded";
39         return 0;
40     }
41
42     TAverageValue averageValue = (TAverageValue)GetProcAddress(hDLL, "AverageValue");
43     if (!averageValue)
44     {
45         cout << "Function AverageValue not founded";
46         return 0;
47     }
48
49     // Тестування функції FindMinimum
50     int arrFindMin[5] = { 5, -3, 8, 1, 0 };
51     int minResult = findMinimum(arrFindMin, sizeof(arrFindMin));
52     assert(minResult == -3);
53     cout << "FindMinimum function test passed." << endl;
54
55
56     // Тестування функції CalculateFactorial
57     // Тест 1: Позитивне число
58     int n1 = 5;
59     int expected1 = 120;
60     assert(calculateFactorial(n1) == expected1);
61
62     // Тест 2: Нуль
63     int n2 = 0;
64     int expected2 = 1;
65     assert(calculateFactorial(n2) == expected2);
66
67     // Тест 3: Від'ємне число
68     int n3 = -3;
69     int expected3 = -1;
70     assert(calculateFactorial(n3) == expected3);
71     cout << "CalculateFactorial tests passed." << endl;
72
73
74     // Тестування функції ContainsSub
75     char toCheck[] = "Hello, World!";
76     char toFind[] = "World";
77     bool containsResult = containsSub(toCheck, sizeof(toCheck), toFind, sizeof(toFind), true);
78     assert(containsResult);
79 }
```

```

79
80 char toCheck2[] = "Hello, World!";
81 char toFind2[] = "world"; // Маленька 'w', оскільки регістр не враховується
82 bool containsResult2 = containsSub(toCheck, sizeof(toCheck), toFind, sizeof(toFind), false);
83 assert(containsResult2);
84 cout << "ContainsSub function tests passed." << endl;
85
86
87 // Тестування функції AverageValue
88 int numArr[5] = { 3, 5, 7, 10, 2 }; // Average = 5.4
89 double averageResult = averageValue(numArr, sizeof(numArr));
90 assert(averageResult == 5.4);
91 cout << "AverageValue function test passed." << endl;
92
93 FreeLibrary(hDLL);
94
95 return 0;
96

```

Після запуску все пройшло успішно.

```

Microsoft Visual Studio Debug Console
FindMinimum function test passed.
CalculateFactorial tests passed.
ContainsSub function tests passed.
AverageValue function test passed.

```

Додала функцію для тимчасового призупинення виконання програми, адже тепер середовище не зможе зупиняти процес за користувача.

```

cout << "Press any key to continue . . . ";
cin.get();

```

Сам \*.dll файл не додавала у папку програми, адже він знаходиться у системній папці Windows і тому перша програма сама зможе його знайти, а у другій програмі і так явно вказала шлях до файлу.

Потім запустила програми з папки Release та перевірила, що все працює.

1 програма:

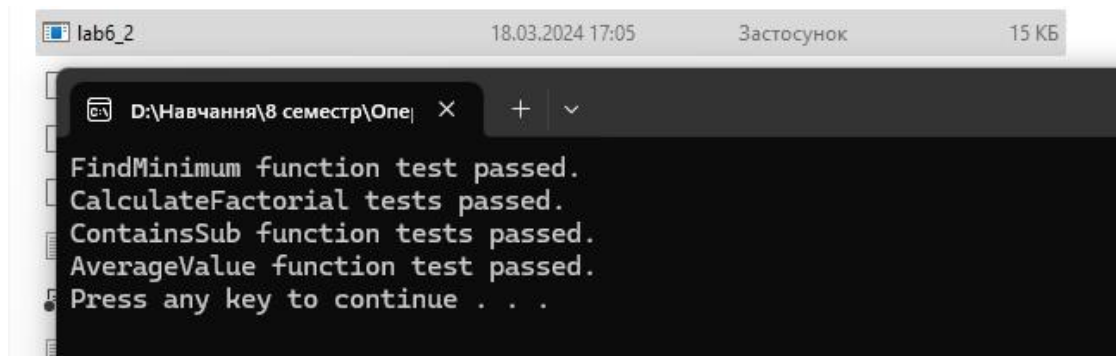
```

lab6 18.03.2024 17:08 Застосунок 14 КБ
D:\Навчання\8 семестр\One
FindMinimum function test passed.
CalculateFactorial tests passed.
ContainsSub function tests passed.
AverageValue function test passed.
Press any key to continue . . .

```



2 програма:



The screenshot shows a Windows application window titled "lab6\_2" with a timestamp of "18.03.2024 17:05" and a file size of "15 KB". The application is running a test suite, and the output is displayed in a dark-themed console window. The output shows that all tests passed, including "FindMinimum function test", "CalculateFactorial tests", "ContainsSub function tests", and "AverageValue function test". The console window also shows a prompt to "Press any key to continue . . .".

```
FindMinimum function test passed.  
CalculateFactorial tests passed.  
ContainsSub function tests passed.  
AverageValue function test passed.  
Press any key to continue . . .
```