

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА
ФРАНКА

Факультет прикладної математики та інформатики

Кафедра програмування

Лабораторна робота № 5

Створення DLL

Виконала
студентка групи ПМО-41
Кравець Ольга

2024

Хід роботи

1. Склала функції:

- ✓ Функція знаходження мінімального елементу.
- ✓ Функція для обчислення факторіалу числа.
- ✓ Функція, яка перевіряє, чи містить заданий `char*` певний підрядок `char*`, з можливістю ввімкнення або вимкнення чутливості до регістру символів за допомогою параметру `bool`.
- ✓ Функція знаходження середнього значення.

2. Створила проєкт в Visual Studio, де реалізувала ці функції і потім їх тестувала.

Код функцій:

```
Source.cpp  X
Lab5DLL  (Global Scope)  AverageValue(int * numArr, int
1  #include <iostream>
2  #include <cassert>
3  #include <string>
4
5  using namespace std;
6
7  int FindMinimum(int* s, int arrSize);
8  int CalculateFactorial(int n);
9  bool ContainsSub(char* toCheck, int checkedSize, char* toFind, int foundSize, bool caseSensitive);
10 double AverageValue(int* numArr, int arrSize);
11
```

```
// Функція знаходження мінімального елементу
int FindMinimum(int* s, int arrSize)
{
    if (arrSize == 0)
    {
        return 0;
    }

    int min = s[0];

    for (int i = 1; i < arrSize / sizeof(min); ++i)
    {
        if (s[i] < min)
        {
            min = s[i];
        }
    }

    return min;
}
```

```
// Функція для обчислення факторіалу числа.
```

```
int CalculateFactorial(int n)
{
    if (n < 0) {
        return -1;
    }
    else if (n == 0 || n == 1) {
        return 1;
    }
    else {
        int factorial = 1;
        for (int i = 2; i <= n; ++i) {
            factorial *= i;
        }
        return factorial;
    }
}
```

```
// Функція, яка перевіряє, чи містить заданий char* певний підрядок char*,
// з можливістю ввімкнення або вимкнення чутливості до регістру символів за допомогою параметру bool.
bool ContainsSub(char* toCheck, int checkedSize, char* toFind, int foundSize, bool caseSensitive)
```

```
{
    int checkedLen = checkedSize - 1;
    int foundLen = foundSize - 1;

    for (int i = 0; i <= checkedLen - foundLen; ++i)
    {
        if (checkedLen < foundLen + i)
        {
            return false;
        }

        bool result = true;

        for (int j = 0; j < foundLen; ++j)
        {
            char left = toCheck[i + j];
            char right = toFind[j];

            if (!caseSensitive)
            {
                left = tolower(left);
                right = tolower(right);
            }

            if (left != right)
            {
                result = false;
                break;
            }
        }

        if (result)
        {
            return true;
        }
    }

    return false;
}
```

```

// Функція знаходження середнього значення
double AverageValue(int* numArr, int arrSize)
{
    if (arrSize == 0)
    {
        return 0;
    }

    int arrLen = arrSize / sizeof(*numArr);
    double sum = 0;

    for (int i = 0; i < arrLen; ++i)
    {
        sum += numArr[i];
    }

    return sum / arrLen;
}

```

Тестування функцій:

```

// Тестування функції FindMinimum
int arrFindMin[5] = { 5, -3, 8, 1, 0 };
int minResult = FindMinimum(arrFindMin, sizeof(arrFindMin));
assert(minResult == -3);
cout << "FindMinimum function test passed." << endl;

// Тестування функції CalculateFactorial
// Тест 1: Позитивне число
int n1 = 5;
int expected1 = 120;
assert(CalculateFactorial(n1) == expected1);

// Тест 2: Нуль
int n2 = 0;
int expected2 = 1;
assert(CalculateFactorial(n2) == expected2);

// Тест 3: Від'ємне число
int n3 = -3;
int expected3 = -1;
assert(CalculateFactorial(n3) == expected3);
cout << "CalculateFactorial tests passed." << endl;

```

```

// Тестування функції ContainsSub
char toCheck[] = "Hello, World!";
char toFind[] = "World";
bool containsResult = ContainsSub(toCheck, sizeof(toCheck), toFind, sizeof(toFind), true);
assert(containsResult);

char toCheck2[] = "Hello, World!";
char toFind2[] = "world"; // Маленька 'w', оскільки регістр не враховується
bool containsResult2 = ContainsSub(toCheck, sizeof(toCheck), toFind, sizeof(toFind), false);
assert(containsResult2);
cout << "ContainsSub function tests passed." << endl;

```

```
// Тестування функції AverageValue
int numArr[5] = { 3, 5, 7, 10, 2 }; // Average = 5.4
double averageResult = AverageValue(numArr, sizeof(numArr));
assert(averageResult == 5.4);
cout << "AverageValue function test passed." << endl;
```

Усі тести пройшли успішно.

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

- Створила другий проєкт такого самого типу з тими самими функціями, але тепер параметри функцій мають типи Windows. При цьому одночасно замінила типи деяких локальних змінних функцій.

```
Source.cpp
Lab5DLL2 (Global Scope) FindMinimum(LPINT s, INT arrSize)
1 #include <iostream>
2 #include <cassert>
3 #include <string>
4 #include <windows.h>
5
6 using namespace std;
7
8 INT FindMinimum(LPINT s, INT arrSize);
9 INT CalculateFactorial(INT n);
10 BOOL ContainsSub(LPSTR toCheck, INT checkedSize, LPSTR toFind, INT foundSize, BOOL caseSensitive);
11 DOUBLE AverageValue(LPINT numArr, INT arrSize);
12
```

Функції:

```
// Функція знаходження мінімального елементу
INT FindMinimum(LPINT s, INT arrSize)
{
    if (arrSize == 0)
    {
        return 0;
    }

    INT min = s[0];

    for (INT i = 1; i < arrSize / sizeof(min); ++i)
    {
        if (s[i] < min)
        {
            min = s[i];
        }
    }

    return min;
}
```

```

// Функція для обчислення факторіалу числа.
INT CalculateFactorial(INT n)
{
    if (n < 0) {
        return -1;
    }
    else if (n == 0 || n == 1) {
        return 1;
    }
    else {
        int factorial = 1;
        for (int i = 2; i <= n; ++i) {
            factorial *= i;
        }
        return factorial;
    }
}

```

```

// Функція, яка перевіряє, чи містить заданий LPSTR певний підрядок LPSTR,
// з можливістю ввімкнення або вимкнення чутливості до регістру символів за допомогою параметру BOOL.
BOOL ContainsSub(LPSTR toCheck, INT checkedSize, LPSTR toFind, INT foundSize, BOOL caseSensitive)
{
    INT checkedLen = checkedSize - 1;
    INT foundLen = foundSize - 1;

    for (INT i = 0; i <= checkedLen - foundLen; ++i)
    {
        if (checkedLen < foundLen + i)
        {
            return false;
        }

        BOOL result = true;

        for (INT j = 0; j < foundLen; ++j)
        {
            CHAR left = toCheck[i + j];
            CHAR right = toFind[j];

            if (!caseSensitive)
            {
                left = tolower(left);
                right = tolower(right);
            }

            if (left != right)
            {
                result = false;
                break;
            }
        }

        if (result)
        {
            return true;
        }
    }

    return false;
}

```



```

// Функція знаходження середнього значення
DOUBLE AverageValue(LPINT numArr, INT arrSize)
{
    if (arrSize == 0)
    {
        return 0;
    }

    INT arrLen = arrSize / sizeof(*numArr);
    DOUBLE sum = 0;

    for (INT i = 0; i < arrLen; ++i)
    {
        sum += numArr[i];
    }

    return sum / arrLen;
}

```

Результати тестування співпали з кроком 2.

```

FindMinimum function test passed.
CalculateFactorial tests passed.
ContainsSub function tests passed.
AverageValue function test passed.

```

4. Створила новий DLL проєкт і перенесла туди функції, написала їх прототипи та експортувала.

`#include "WTypesbase.h"` - включила у файл, бо інакше не знаходило тип Windows DOUBLE.

```

MyLib (Global Scope) FindMinimum(LPINT s, INT arrSize)
1 // dllmain.cpp : Defines the entry point for the DLL application.
2 #include "framework.h"
3 #include "WTypesbase.h"
4
5 extern "C" __declspec(dllexport) INT(__stdcall FindMinimum) (LPINT s, INT arrSize);
6 extern "C" __declspec(dllexport) INT(__stdcall CalculateFactorial) (INT n);
7 extern "C" __declspec(dllexport) BOOL ContainsSub(LPSTR toCheck, INT checkedSize, LPSTR toFind, INT foundSize, BOOL caseSen:
8 extern "C" __declspec(dllexport) DOUBLE AverageValue(LPINT numArr, INT arrSize);
9

```

```

9
10 // Функція знаходження мінімального елементу
11 INT FindMinimum(LPINT s, INT arrSize)
12 {
13     if (arrSize == 0)
14     {
15         return 0;
16     }
17
18     INT min = s[0];
19
20     for (INT i = 1; i < arrSize / sizeof(min); ++i)
21     {
22         if (s[i] < min)
23         {
24             min = s[i];
25         }
26     }
27
28     return min;
29 }

```

```

// Функція для обчислення факторіалу числа.
INT CalculateFactorial(INT n)
{
    if (n < 0) {
        return -1;
    }
    else if (n == 0 || n == 1) {
        return 1;
    }
    else {
        int factorial = 1;
        for (int i = 2; i <= n; ++i) {
            factorial *= i;
        }
        return factorial;
    }
}

```



```

49 // Функція, яка перевіряє, чи містить заданий LPSTR певний підрядок LPSTR,
50 // з можливістю ввімкнення або вимкнення чутливості до регістру символів за допомогою параметру BOOL.
51 BOOL ContainsSub(LPSTR toCheck, INT checkedSize, LPSTR toFind, INT foundSize, BOOL caseSensitive)
52 {
53     INT checkedLen = checkedSize - 1;
54     INT foundLen = foundSize - 1;
55
56     for (INT i = 0; i <= checkedLen - foundLen; ++i)
57     {
58         if (checkedLen < foundLen + i)
59         {
60             return false;
61         }
62
63         BOOL result = true;
64
65         for (INT j = 0; j < foundLen; ++j)
66         {
67             CHAR left = toCheck[i + j];
68             CHAR right = toFind[j];
69
70             if (!caseSensitive)
71             {
72                 left = tolower(left);
73                 right = tolower(right);
74             }
75
76             if (left != right)
77             {
78                 result = false;
79                 break;
80             }
81         }
82         if (result)
83         {
84             return true;
85         }
86     }
87     return false;
88 }
89

```

```

// Функція знаходження середнього значення
DOUBLE AverageValue(LPINT numArr, INT arrSize)
{
    if (arrSize == 0)
    {
        return 0;
    }

    INT arrLen = arrSize / sizeof(*numArr);
    DOUBLE sum = 0;

    for (INT i = 0; i < arrLen; ++i)
    {
        sum += numArr[i];
    }

    return sum / arrLen;
}

```

Забілдила проєкт. Усе пройшло успішно і в результаті в папці Debug цілого проєкту є файли MyLib.dll, MyLib.lib, які будуть потрібні для використання в інших програмах користувачів.

Ім'я	Дата змінення	Тип	Розмір
MyLib.tlog	15.03.2024 22:12	Папка файлів	
dllmain	15.03.2024 22:12	3D Object	33 КБ
MyLib.dll	15.03.2024 22:12	Розширення заст...	59 КБ
MyLib.dll.recipe	15.03.2024 22:12	Файл RECIPE	1 КБ
MyLib.exp	15.03.2024 22:12	Exports Library File	2 КБ
MyLib.ilc	15.03.2024 22:12	Incremental Linke...	507 КБ
MyLib.lib	15.03.2024 22:12	Object File Library	3 КБ
MyLib	15.03.2024 22:12	Текстовий докум...	1 КБ
MyLib.pdb	15.03.2024 22:12	Program Debug D...	852 КБ
MyLib.vcxproj.FileListAbsolute	15.03.2024 22:12	Текстовий докум...	1 КБ
vc143.idb	15.03.2024 22:12	VC++ Minimum R...	155 КБ
vc143.pdb	15.03.2024 22:12	Program Debug D...	84 КБ

Тому їх скопіювала в окрему від проєкту папку, до якої будуть мати доступ прикладні програми користувачів.

Ім'я	Дата змінення	Тип	Розмір
MyLib.dll	15.03.2024 22:12	Розширення заст...	59 КБ
MyLib.lib	15.03.2024 22:12	Object File Library	3 КБ