

Лабораторна робота № 5

Виконав:

студент ПМІ-44

Копина Ілля

Тема: Побудова бібліотеки DLL мовою C++

Планування Завдання:

Для створення DLL були придумані 5 функцій, які оперують стандартними типами даних для повернення результату та для приймання параметрів. Усі функції працюють з рядками та можуть допомогти користувачу з написанням великих програм, які працюють з текстом. Ось перелік функцій:

- Функція для знаходження найбільш повторюваного символу у рядку.
- Функція для перевірки рядка на наявність властивості паліндрому.
- Функція для знаходження індексу під рядка у рядку.
- Функція для знаходження унікальних символів у рядку.
- Функція для перетворення рядка у Title формат.

Хід Роботи:

Був створений Пустий Проект з назвою **LabDll**, після чого був доданий файл **Source.cpp** у якому пізніше будуть зберігатися функції та тести для них.

Тести для функцій представляють собою простий виклик функцій і відправку результатів на консоль разом з значеннями, які мають бути повернені. Таким чином буде легко перевірити, що програма справді працює і не потрібно буде тратити часу на справжнє тестування за допомогою юніт тестів. Як можна побачити з зображення, прототипи функцій знаходяться на початку файлу, а самі тести безпосередньо в функцію **Main**. Окрім цього, була створена допоміжна функція для обрахунку розміру елементів у динамічному масиві символів. Код самих функцій є доданий вкінці звіту

```
1 #include <iostream>
2
3 using namespace std;
4
5 char findTheMostUsedChar(char* text);
6 bool isPalindrome(char* text);
7 int findSubstringIndex(char* text, char* substring);
8 char* findUniqueChars(char* text);
9 char* makeTitleString(char* text);
10
11
12 int main()
13 {
14     char test[] = "Hey, hello there";
15     char substring[] = "the";
16     char testPolindrom[] = "racecar";
17
18     cout << findTheMostUsedChar(test) << " | e" << endl;
19     cout << isPalindrome(testPolindrom) << " | 1" << endl;
20     cout << findSubstringIndex(test, substring) << " | 11" << endl;
21
22     char* uniqueChars = findUniqueChars(test);
23
24     for (int i = 0; i < 5; i++)
25     {
26         cout << uniqueChars[i];
27     }
28
29     cout << " | Y,OTR" << endl;
30
31     char* titleString = makeTitleString(test);
32
33     for (int i = 0; i < 16; i++)
34     {
35         cout << titleString[i];
36     }
37     cout << " | Hey, Hello There" << endl;
38 }
39
```

Microsoft Visual Studio Debug Console

```
e | e
1 | 1
11 | 11
Y,OTR | Y,OTR
Hey, Hello There | Hey, Hello There

C:\Users\Admin\source\repos\LabDll\Debug\La
To automatically close the console when de
le when debugging stops.
Press any key to close this window . . .
```

Наступним етапом було створення ще одного проекту з тими самими функціями, але з параметрами функцій типу **Windows**. Для цього я ознайомився з довідкою системи про співставлення типів **Windows** і стандартних типів **C++**. Після перенесення функцій та перевірки тестів, усе було готово до створення DLL.

```
1 #include <iostream>
2 #include <Windows.h>
3
4 using namespace std;
5
6 CHAR findTheMostUsedChar(LPSTR text);
7 BOOL isPalindrome(LPSTR text);
8 INT findSubstringIndex(LPSTR text, LPSTR substring);
9 LPSTR findUniqueChars(LPSTR text);
10 LPSTR makeTitleString(LPSTR text);
11
12
13 int main()
14 {
15     char test[] = "Hey, hello there";
16     char substring[] = "the";
17     char testPolindrom[] = "racecar";
18
19     cout << findTheMostUsedChar(test) << " | e" << endl;
20     cout << isPalindrome(testPolindrom) << " | 1" << endl;
21     cout << findSubstringIndex(test, substring) << " | 11" << endl;
22
23     char* uniqueChars = findUniqueChars(test);
24
25     for (int i = 0; i < 5; i++)
26     {
27         cout << uniqueChars[i];
28     }
29
30     cout << " | Y,OTR" << endl;
31
32     char* titleString = makeTitleString(test);
33
34     for (int i = 0; i < 16; i++)
35     {
36         cout << titleString[i];
37     }
38     cout << " | Hey, Hello There" << endl;
39 }
40
```

Microsoft Visual Studio Debug Console

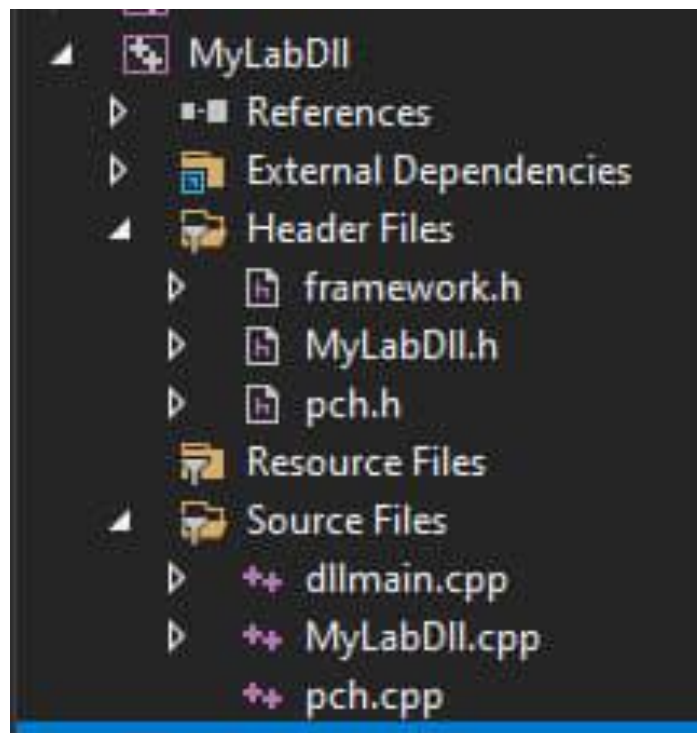
```
e | e
1 | 1
11 | 11
Y,OTR | Y,OTR
Hey, Hello There | Hey, Hello There

C:\Users\Admin\source\repos\LabDll\Debug\W
To automatically close the console when de
le when debugging stops.
Press any key to close this window . . .
```

Тому я створив новий **Dynamic Link Library** проект і слідуючи інструкції з офіційного сайту переніс свої функції у новий проект. А саме: були створені **MyLabDll.cpp** та **MyLabDll.h** файли куди перенесли реалізацію функцій та прототипи відповідно. Також файл з прототипами функцій був модифікований за інструкцією та набув такого вигляду:

```
1  #pragma once
2
3  #ifdef MYLABDLL_EXPORTS
4  #define MYLABDLL_API __declspec(dllexport)
5  #else
6  #define MYLABDLL_API __declspec(dllimport)
7  #endif
8
9  #include <windows.h>
10
11  extern "C" MYLABDLL_API CHAR findTheMostUsedChar(LPSTR text);
12  extern "C" MYLABDLL_API BOOL isPalindrome(LPSTR text);
13  extern "C" MYLABDLL_API INT findSubstringIndex(LPSTR text, LPSTR substring);
14  extern "C" MYLABDLL_API LPSTR findUniqueChars(LPSTR text);
15  extern "C" MYLABDLL_API LPSTR makeTitleString(LPSTR text);
```

Ось як виглядала структура готового проекту:







Результат Роботи:

Після запуску програми була створена Debug папка у якій збереглися потрібні нам файли.

```
Output  Source.cpp  MyLabDll.cpp  pch.h  framework.h  MyLabDll.h  pch.cpp  dllmain.cpp
Show output from: Build
Build started...
1>----- Build started: Project: MyLabDll, Configuration: Debug Win32 -----
1>LINK : C:\Users\Admin\source\repos\LabDll\Debug\MyLabDll.dll not found or not built by the last incremental link; performing full link
1>  Creating library C:\Users\Admin\source\repos\LabDll\Debug\MyLabDll.lib and object C:\Users\Admin\source\repos\LabDll\Debug\MyLabDll.exp
1>MyLabDll.vcxproj -> C:\Users\Admin\source\repos\LabDll\Debug\MyLabDll.dll
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

» source » repos » LabDll » Debug

Name	Date modified	Type	Size
 MyLabDll.dll	5/9/2023 4:45 AM	Application exten...	63 KB
 MyLabDll.exp	5/9/2023 4:45 AM	Exports Library File	2 KB
 MyLabDll.lib	5/9/2023 4:45 AM	Object File Library	3 KB
 MyLabDll.pdb	5/9/2023 4:45 AM	Program Debug D...	1,492 KB

Код Программы:

```
LabDll (Global Scope)
1  #include <iostream>
2
3  using namespace std;
4
5  char findTheMostUsedChar(char* text);
6  bool isPalindrome(char* text);
7  int findSubstringIndex(char* text, char* substring);
8  char* findUniqueChars(char* text);
9  char* makeTitleString(char* text);
10
11
12  int main()
13  {
14      char test[] = "Hey, hello there";
15      char substring[] = "the";
16      char testPolindrom[] = "racecar";
17
18      cout << findTheMostUsedChar(test) << " | e" << endl;
19      cout << isPalindrome(testPolindrom) << " | 1" << endl;
20      cout << findSubstringIndex(test, substring) << " | 11" << endl;
21
22      char* uniqueChars = findUniqueChars(test);
23
24      for (int i = 0; i < 5; i++)
25      {
26          cout << uniqueChars[i];
27      }
28
29      cout << " | Y,OTR" << endl;
30
31      char* titleString = makeTitleString(test);
32
33      for (int i = 0; i < 16; i++)
34      {
35          cout << titleString[i];
36      }
37      cout << " | Hey, Hello There" << endl;
38  }
39
40  int getSize(char* str)
41  {
42      int len = 0;
43      while (*str != '\0')
44      {
45          len++;
46          str++;
47      }
48
49      return len;
50  }
51
52  char findTheMostUsedChar(char* text)
53  {
54      int size = getSize(text);
```



```

54     int size = getSize(text);
55     char resultChar = ' ';
56     int charCount = 0;
57
58     for (int i = 0; i < size; i++)
59     {
60         int tempCount = 0;
61         for (int j = i; j < size; j++)
62         {
63             if (text[i] == text[j])
64             {
65                 tempCount++;
66             }
67         }
68
69         if (tempCount > charCount)
70         {
71             charCount = tempCount;
72             resultChar = text[i];
73         }
74     }
75
76     return resultChar;
77 }
78
79 bool isPalindrome(char* text)
80 {
81     int size = getSize(text);
82
83     for (int i = 0; i < size / 2; i++)
84     {
85         if (text[i] != text[size - i - 1])
86         {
87             return false;
88         }
89     }
90
91     return true;
92 }
93
94 int findSubstringIndex(char* text, char* substring)
95 {
96     int textSize = getSize(text);
97     int substringSize = getSize(substring);
98
99     for (int i = 0; i <= textSize - substringSize; i++)
100     {
101         bool found = true;
102         for (int j = 0; j < substringSize; j++)
103         {
104             if (text[i + j] != substring[j])
105             {

```

```

105         {
106             found = false;
107             break;
108         }
109     }
110
111     if (found)
112     {
113         return i;
114     }
115 }
116
117 return -1;
118 }
119
120 char* findUniqueChars(char* text)
121 {
122     int size = getSize(text);
123     string temp = "";
124
125     for (int i = 0; i < size; i++)
126     {
127         bool found = false;
128         for (int j = 0; j < size; j++)
129         {
130             if (i != j && toupper(text[i]) == toupper(text[j]))
131             {
132                 found = true;
133                 break;
134             }
135         }
136
137         if (!found)
138         {
139             temp += toupper(text[i]);
140         }
141     }
142
143     int tempSize = temp.length();
144     char* result = new char[tempSize];
145
146     for (int i = 0; i < tempSize; i++)
147     {
148         result[i] = temp[i];
149     }
150
151     return result;
152 }
153
154 char* makeTitleString(char* text)
155 {
156     int size = getSize(text);

```

```

156     int size = getSize(text);
157     string temp = "";
158
159     bool isNewWord = true;
160     for (int i = 0; i < size; i++)
161     {
162         if (isNewWord && isalpha(text[i]))
163         {
164             temp += toupper(text[i]);
165             isNewWord = false;
166         }
167         else if (text[i] == ' ' || ispunct(text[i]))
168         {
169             temp += text[i];
170             isNewWord = true;
171         }
172         else
173         {
174             temp += text[i];
175         }
176     }
177
178     int tempSize = temp.length();
179     char* result = new char[tempSize];
180
181     for (int i = 0; i < tempSize; i++)
182     {
183         result[i] = temp[i];
184     }
185
186     return result;
187 }

```