

## ЛАБОРАТОРНА РОБОТА №5

**Тема:** Динамічно зв'язувані бібліотеки.

**Мета роботи:** Вивчення роботи з DLL

### Завдання

1. Написати програмний продукт, який буде мати об'єктно-орієнтовану архітектуру. Обов'язкове створення та реалізація двох класів. Перший - інтерфейс, другий - обчислення.

2. Кожен клас має розташовуватися в окремій DLL.

Лістинг файлу interface.h:

```
#pragma once

class __declspec(dllexport) ICalculator
{
public:
    virtual void sort_array(int* array, int length) = 0;
    virtual ~ICalculator() {}
};
```

Лістинг файлу calculator.h:

```
#pragma once

#include "../interface/interface.h"

class __declspec(dllexport) Calculator : public ICalculator
{
public:
    Calculator();
    ~Calculator();
    static void sort_chunk(int* array, size_t start, size_t end);
    void sort_array(int* array, int length);
    static int compare(const void* a, const void* b);
};
```

					ДУ«Житомирська політехніка».25.121.27.000 – Лр5			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Семенчук О.А.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Власенко О.В.						1
Керівник								9
Н. контр.							ФІКТ Гр. ІПЗ-23-1[2]	
Зав. каф.								

```
extern "C" __declspec(dllexport) ICalculator* CreateCalculator();
```

### Лістинг файлу calculator.cpp:

```
#include "../calculator.h"
#include <algorithm>
#include <iostream>
#include <thread>
#include <vector>

Calculator::Calculator()
{
    std::cout << "Calculator instance constructed." << std::endl;
}

Calculator::~~Calculator()
{
    std::cout << "Calculator instance deconstructed!" << std::endl;
}

void Calculator::sort_chunk(int* array, size_t start, size_t end)
{
    qsort(array + start, end - start, sizeof(int), compare);
}

int Calculator::compare(const void* a, const void* b)
{
    return (*((const int*)a) - *((const int*)b));
}

void Calculator::sort_array(int* array, int length)
{
    std::cout << "Sorting array..." << std::endl;

    int number_of_threads = std::thread::hardware_concurrency();

    std::vector<std::thread> threads;
    size_t chunk_size = length / number_of_threads;

    for (int i = 0; i < number_of_threads; i++)
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр5	Арк.
		Власенко О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

{
    size_t start = i * chunk_size;
    size_t end = (i == chunk_size - 1) ? length : start + chunk_size;

    threads.emplace_back(sort_chunk, array, start, end);
}

for (int i = 0; i < threads.size(); i++) threads[i].join();

size_t current_chunk_size = chunk_size;

for (int i = 1; i < number_of_threads; ++i)
{
    size_t middle = i * chunk_size;
    size_t end = (i == number_of_threads - 1) ? length : (i + 1) *
chunk_size;
    std::inplace_merge(array, array + middle - chunk_size, array +
end);
}
}

extern "C" __declspec(dllexport) ICalculator* CreateCalculator()
{
    return new Calculator();
}

```

3. DLL яка реалізує інтерфейс має завантажуватися разом з основною програмою.

```

PS C:\Users\357\Desktop\projects\system-network-programming> cl main.cpp /link .\interface\interface.lib
Microsoft (R) C/C++ Optimizing Compiler Version 19.40.33812 for x86
Copyright (C) Microsoft Corporation. All rights reserved.

main.cpp
C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\MSVC\14.40.33807\include\ostream(425): warning C4530: C++ exception handler used, but unwind semantics are not enabled. Specify /EHsc
C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\MSVC\14.40.33807\include\ostream(425): note: the template instantiation context (the oldest one first) is
main.cpp(34): note: see reference to class template instantiation 'std::basic_ostream<char,std::char_traits<char>,std::allocator<char>>' being compiled
C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\MSVC\14.40.33807\include\sstream(727): note: see reference to class template instantiation 'std::basic_ostream<char,std::char_traits<char>>' being compiled
C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\MSVC\14.40.33807\include\ostream(418): note: while compiling class template member function 'std::basic_ostream<char,std::char_traits<char>> &std::basic_ostream<char,std::char_traits<char>>::operator <<(double)'
main.cpp(35): note: see the first reference to 'std::basic_ostream<char,std::char_traits<char>>::operator <<' in 'get_human_readable_bytes'
Microsoft (R) Incremental Linker Version 14.40.33812.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:main.exe
.\interface\interface.lib
main.obj
Creating library main.lib and object main.exp

```

Рис. 1 Компіляція main.cpp

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр5	Арк.
		Власенко О.В				3
Змн.	Арк.	№ докум.	Підпис	Дата		

Під час компіляції програми відбувається зв'язування основного файлу main.cpp з бібліотекою interface.lib.

4. DLL яка реалізує обчислення має завантажуватися пізніше, вже під час роботи програмного продукту, за вимогою користувача. А також після використання відвантажуватися з пам'яті. Перед завантаженням необхідно перевірити чи є вже бібліотека у пам'яті.

Фрагменти коду:

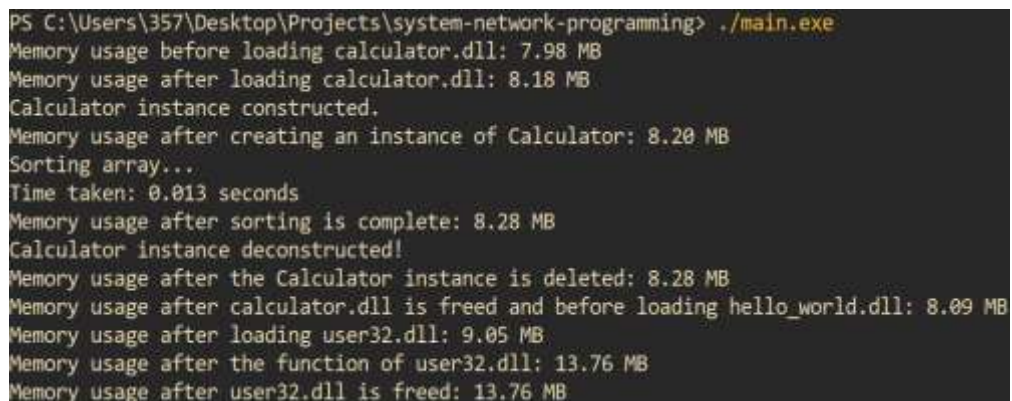
```
HMODULE module = GetModuleHandle(calculator_library_path.c_str());  
if (!module) module = LoadLibrary(calculator_library_path.c_str());  
  
delete calculator;  
FreeLibrary(module);
```

5. Підключити та використати будь які функції для прикладу, будь яку не власну бібліотеку.

Фрагменти коду:

```
HMODULE external_module = LoadLibrary(external_module_name.c_str());  
  
external_function external = (exter  
nal_function)GetProcAddress(external_module, "MessageBoxA");  
  
external(NULL, "user32.dll", "user32.dll", MB_OK);
```

6. При реалізації класу обчислень, використати алгоритм повного перебору, або сортування qsort, та використати породження багатьох потоків. Вивчити поведінку системи при граничному використанні пам'яті та породжених потоків. Показати графіки використання ресурсів. Проаналізувати залежності в них.



```
PS C:\Users\357\Desktop\Projects\system-network-programming> ./main.exe  
Memory usage before loading calculator.dll: 7.98 MB  
Memory usage after loading calculator.dll: 8.18 MB  
Calculator instance constructed.  
Memory usage after creating an instance of Calculator: 8.20 MB  
Sorting array...  
Time taken: 0.013 seconds  
Memory usage after sorting is complete: 8.28 MB  
Calculator instance deconstructed!  
Memory usage after the Calculator instance is deleted: 8.28 MB  
Memory usage after calculator.dll is freed and before loading hello_world.dll: 8.09 MB  
Memory usage after loading user32.dll: 9.05 MB  
Memory usage after the function of user32.dll: 13.76 MB  
Memory usage after user32.dll is freed: 13.76 MB
```

Рис. 2 Запуск програми

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр5	Арк.
		Власенко О.В				4
Змн.	Арк.	№ докум.	Підпис	Дата		

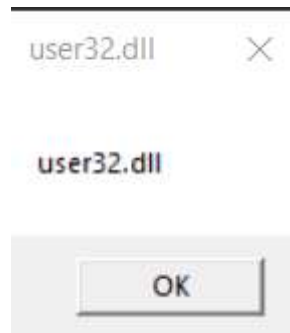


Рис. 3 Функція MessageBoxA бібліотеки user32.dll

### Лістинг файлу main.cpp:

```
#include <iomanip>
#include <iostream>
#include <random>
#include <chrono>
#include <sstream>
#include <string>
#include <windows.h>
#include <Psapi.h>
#include "../interface/interface.h"

typedef ICalculator* (*create_calculator_function)();
typedef void (*external_function)(HWND, LPCSTR, LPCSTR, UINT);

int get_memory_usage()
{
    PROCESS_MEMORY_COUNTERS process_memory;
    GetProcessMemoryInfo(GetCurrentProcess(), &process_memory,
        sizeof(process_memory));

    return process_memory.WorkingSetSize;
}

std::string get_human_readable_bytes(int bytes)
{
    const char* sizes[] = { "B", "KB", "MB", "GB", "TB", "PB" };
    int order = 0;
    double readable_size = static_cast<double>(bytes);

    while (readable_size >= 1024 && order < 5)
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр5	Арк.
		Власенко О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		5

```

    {
        order++;
        readable_size /= 1024;
    }

    std::ostringstream oss;
    oss << std::fixed << std::setprecision(2) << readable_size << " " <<
    sizes[order];

    return oss.str();
}

void display_memory_info(std::string message="")
{
    std::string default_message = "Memory usage";

    if(message == "") message = default_message;

    std::cout << message << ": " <<
        get_human_readable_bytes(get_memory_usage()) << std::endl;
}

void output_array(int* array, int length)
{
    for (int i = 0; i < length; i++)
    {
        std::cout << array[i];

        if (i + 1 != length) std::cout << " ";
    }

    std::cout << std::endl;
}

int get_random_integer(int min, int max)
{
    std::random_device rd;
    std::uniform_int_distribution<> dist(min, max);

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр5	Арк.
		Власенко О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

        return dist(rd);
    }

int main()
{
    std::string calculator_library_path= "./calculator/calculator.dll";

    int length = 1000000, min = -10, max = 10;
    int* array = new int[length];

    for (int i = 0; i < length; i++) array[i] = get_random_integer(min,
max);

    display_memory_info("Memory usage before loading calculator.dll");

    HMODULE module = GetModuleHandle(calculator_library_path.c_str());
    if (!module) module = LoadLibrary(calculator_library_path.c_str());

    std::cout << "Memory usage after loading calculator.dll: " <<
        get_human_readable_bytes(get_memory_usage()) << std::endl;;

    if (!module)
    {
        std::cerr << "Loading calculator library failed with error code: "
<<
        GetLastError() << std::endl;

        return 1;
    }

    create_calculator_function create_calculator =
        (create_calculator_function)GetProcAddress(module, "CreateCalcula-
tor");

    if (!create_calculator)
    {
        std::cerr << std::endl << "Failed to get a create calculator func-
tion!" <<
        " Error code: " << GetLastError() << std::endl;
    }
}

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр5	Арк.
		Власенко О.В				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return 1;
    }

    ICalculator* calculator = create_calculator();

    display_memory_info("Memory usage after creating an instance of Calculator");

    auto start = std::chrono::high_resolution_clock::now();

    calculator->sort_array(array, length);

    auto end = std::chrono::high_resolution_clock::now();

    auto duration =
std::chrono::duration_cast<std::chrono::milliseconds>(end - start);
    std::cout << "Time taken: " << duration.count() / 1000.0 << " seconds"
<< std::endl;

    display_memory_info("Memory usage after sorting is complete");

    delete calculator;

    display_memory_info("Memory usage after the Calculator instance is deleted");

    FreeLibrary(module);

    display_memory_info("Memory usage after calculator.dll is freed and
before loading hello_world.dll");

    std::string external_module_name = "user32.dll";

    HMODULE external_module = LoadLibrary(external_module_name.c_str());

    if (!external_module)
    {
        std::cerr << std::endl << "Loading " << external_module_name <<
        " failed with error code: " << GetLastError() << std::endl;

        return 1;
    }

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр5	Арк.
		Власенко О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		8



```

    }

    display_memory_info("Memory usage after loading " + external_module_name);

    external_function external = (external_function)GetProcAddress(external_module, "MessageBoxA");

    if (!external)
    {
        std::cerr << std::endl << "Could not find the function in " <<
            external_module_name << "." << " Error code: " << GetLastError() << std::endl;

        return 1;
    }

    external(NULL, "user32.dll", "user32.dll", MB_OK);

    display_memory_info("Memory usage after the function of " + external_module_name);

    FreeLibrary(external_module);

    display_memory_info("Memory usage after " + external_module_name + " is freed");

    return 0;
}

```

**Висновок:** Дана робота дозволила ознайомитися з принципами створення та використання динамічно зв'язуваних бібліотек (DLL). У результаті виконання завдання було засвоєно основи модульного програмування та повторного використання коду.

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр5	Арк.
		Власенко О.В				9
Змн.	Арк.	№ докум.	Підпис	Дата		