МIНIСТЕРСТВО ОСВIТИ I НАУКИ УКРАЇНИ

НАЦIОНАЛЬНИЙ ТЕХНIЧНИЙ УНIВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛIТЕХНIЧНИЙ IНСТИТУТ»

Кафедра прикладної математики

ПРОГРАМУВАННЯ ТА ВИКОРИСТАННЯ DLL В ОС WINDOWS

лабораторна робота №3 з дисциплiни:

«Операцiйнi системи»

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-63 | *асистент Громова В. В.* |
| *Артеменко Я.К.* |  |

Київ — 2019

# **ЗМІСТ**

[**ЗМІСТ** 1](#_Toc3730304)

[**Вступ** 2](#_Toc3730305)

[**1 ПОСТАНОВКА ЗАДАЧІ** 3](#_Toc3730306)

[**2** **ОПИС ПРОГРАМИ** 5](#_Toc3730307)

[**ВИСНОВКИ** 6](#_Toc3730308)

[**Додаток А (скріншоти результату)** 7](#_Toc3730309)

[**Додаток Б (текст програми)** 8](#_Toc3730310)

# **Вступ**

Особливiстю операцiйної системи (ОС) Windows є використання бi-

блiотек динамiчного зв’язування (dynamic link librarires, DLL). Бiблiотека

динамiчного зв’язування — це модуль, що мiстить функцiї та данi, якi мо-

жуть використовувати iншi модулi (аплiкацiї чи iншi бiблiотеки). Iншими

словами, DLL — це виконуваний файл, який вiдiграє роль роздiлюваної

бiблiотеки функцiй та даних.

За допомогою динамiчного зв’язування процеси можуть викликати

функцiї, якi не є складовою частиною їхнього коду. Це дозволяє рiзним

аплiкацiям використовувати спiльний код, а також сприяє модулярностi

коду та його розширюваностi. У свою чергу, це спрощує модифiкацiю

функцiональностi аплiкацiй та полегшує повторне використання коду.

ОС Windows сама по собi складається переважно з DLL, i будь-

яка аплiкацiя, яку написано з використанням Windows API, використовує

можливостi DLL.

У данiй роботi розглядатимемо засоби роботи з бiблiотеками дина-

мiчного зв’язування в ОС Windows версiї 2000 та вище.

# **1 ПОСТАНОВКА ЗАДАЧІ**

У данiй лабораторнiй роботi потрiбно ознайомитися з концепцiєю

динамiчного зв’язування в ОС Windows, навчитися створювати за допо-

могою засобiв Windows API бiблiотеки динамiчного зв’язування та вико-

ристувати їх в аплiкацiях.

У рамках виконання лабораторної роботи потрiбно:

а) ознайомитися з теоретичними вiдомостями, викладеними в роз-

дiлi 2;

б) написати будь-якою мовою програмування бiблiотеку динамiчно-

го зв’язування, що реалiзує певнi функцiї; пiд час написання бiблiотеки

не можна використовувати засоби MFC;

в) написати будь-якою мовою програмування програму, яка повинна

демонструвати використання написаної бiблiотеки динамiчного зв’язува-

ння:

1) за допомогою явного зв’язування;

2) за допомогою неявного зв’язування;

програма повинна використовувати тiльки засоби Windows API i не

повинна використовувати засоби MFC;

г) вiдлагодити програму в ОС Windows версiї 2000 та вище;

д) пiдготувати звiт iз лабораторної роботи вiдповiдно до вимог роз-

дiлу 3

# **ОПИС ПРОГРАМИ**

В межах даної лабораторної роботи було розроблено програму, яка використовує створену бібліотеку, яка імітує математичний процес.

*Загальні відомості про програму:*

Програма була написана мовою C з використанням засобів для створення динамічних бібліотек.

Бібліотека має назву «Dll1.dll», яка експортується за допомогою команди extern "C" \_\_declspec(dllexport). Вона виконує імітацію складного математичного процесу та передає результат в головну програму, де вона викликається. Викликається бібліотека за допомогою HMODULE hlib = LoadLibrary(TEXT("Dll1.dll"));

Функції, які використовує програма:

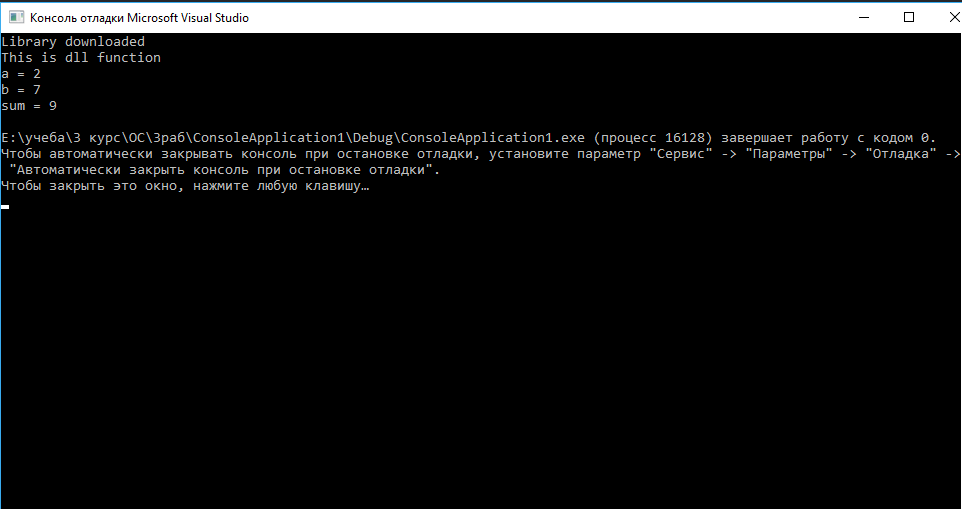
1. **main –** є головною функцією, в якій відбувається перевірка того, чи підключилась бібліотека та виклик функції з біблиотеки.
2. **first\_dll**– функція, яка імітує математичний процес і знаходиться в бібліотечному файлі.

# **ВИСНОВКИ**

На цій лабораторній роботі було здобуто навички по створенню динамічних бібліотек та їх використання в програмуванні. Було створено бібліотеку, в якій імітується математичний процес та програму, яка викликає цю бібліотеку.

# 

# **Додаток А (скріншоти результату)**



# **Додаток Б (текст програми)**

Dll1(Бібліотека)

#include "stdafx.h"

#include <iostream>

using namespace std;

extern "C" \_\_declspec(dllexport)

void first\_dll() {

cout << "This is dll function" << endl;

int a, b, c;

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

c = a + b;

cout << "sum = " << c << endl;

}

ConsoleApplication1(головна програма)

#include "pch.h"

#include <iostream>

#include <conio.h>

#include <Windows.h>

using namespace std;

int main()

{

typedef void(\*f\_class) ();

HMODULE hlib = LoadLibrary(TEXT("Dll1.dll"));

if (hlib == NULL)

{

cout << "Library don`t download\n";

}

if (hlib != NULL)

{

cout << "Library downloaded\n";

}

\_getch();

f\_class first\_dll = (f\_class)GetProcAddress(hlib, "first\_dll");

if (hlib == NULL)

{

cout << "Library don`t download\n";

}

if (hlib != NULL)

{

first\_dll();

}

FreeLibrary(hlib);

return 0;

}