Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет «МЭИ»

Институт информационных и вычислительных технологий

Кафедра Управления и интеллектуальных технологий

**Отчёт по лабораторной работе № 1**

# «C++ и Windows API»

**По курсу «Системное программное обеспечение»**

# Выполнил студент группы А-01-19:

# Китайский А.С.

# Проверил:

# Козлюк Д.А.

# Москва 2022

# Задание на лабораторную работу

Написать программу, использующую Windows API, которая без запросов к пользователю выводит в терминал сведения о системе, а именно:

1. Скорость вычисления квадратного корня при помощи механизма QPC.

//Замеры времени

LARGE\_INTEGER t0;

QueryPerformanceCounter(&t0); //замер текущего времени

double result = sqrt(t0.QuadPart); //вычисление квадратного корня из текущего времени

LARGE\_INTEGER t1;

QueryPerformanceCounter(&t1); //замер текущего времени после вычисления квадратного корня

LARGE\_INTEGER frequency;

QueryPerformanceFrequency(&frequency); //замер частоты счетчика временных интервалов

double ticks = t1.QuadPart - t0.QuadPart; //длительность интервала в тактах

double ticks\_per\_sec = frequency.QuadPart; //отдельная переменная для частоты

double usec\_per\_sec = 1e6; //отдельная переменная для числа микросекунд в секунде

double usec = usec\_per\_sec \* ticks / ticks\_per\_sec; //вычисление времени в микросекундах

printf("result=%g, duration=%.3f usec\n", result, usec);

2. Номер версии Windows.

//Получение версии операционной системы

DWORD info = GetVersion(); //получение номера версии ОС

DWORD mask = 0x0000ffff; //создание маски

DWORD version = info & mask; //наложение маски или побитовое «И»

DWORD platform = info >> 16; //сдвиг вправо на 16 бит для получения старшего слова

DWORD version\_major = version & 0x000000ff; //получение младшего байта младшего слова

DWORD version\_minor = version >> 8; //получение старшего байта младшего слова

DWORD build = 0;

if ((info & 0x10000000) == 0) //проверка равенства старшего бита нулю

{

build = platform >> 8; //получение самого старшего байта

}

cout << "Windows v" << version\_major << "." << version\_minor << " (build " << build << ")" << '\n';

3. Системный каталог, название машины и псевдоним текущего пользователя.

//Получение системного каталога

wchar\_t system\_dir[MAX\_PATH];

if (GetSystemDirectory(system\_dir, MAX\_PATH))

wcout << "System Directory: " << system\_dir << '\n';

else

cout << "Error: " << GetLastError() << '\n';

DWORD buf\_size = MAX\_PATH;

//Получение названия компьютера

wchar\_t comp\_name[MAX\_PATH];

if (GetComputerName(comp\_name, &buf\_size))

wcout << "Computer name: " << comp\_name << '\n';

else

cout << "Error: " << GetLastError() << '\n';

//Получение имени текущего пользователя

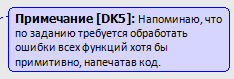
wchar\_t user\_name[MAX\_PATH];

if (GetUserName(user\_name, &buf\_size))

wcout << "User name: " << user\_name << '\n';

else

cout << "Error: " << GetLastError() << '\n';



4. Список логических томов с их названиями, первой точкой монтирования размером и количеством свободного места.

//Получение информации о дисках

wchar\_t buffer[MAX\_PATH];

wchar\_t buf\_path[MAX\_PATH \* 4];

ULARGE\_INTEGER free\_bytes, total\_bytes;

HANDLE search = FindFirstVolume(buffer, sizeof(buffer)); //определение имени первого тома

int i = 0; //инициализация переменной для подсчета номера тома

do {

wcout << "Technical name " << i << " volume:" << buffer << '\n'; //печать техничекого имени i тома

i++;

DWORD buf\_size = MAX\_PATH \* 4;

GetVolumePathNamesForVolumeName(buffer, buf\_path, buf\_size, &buf\_size); //получение из технического имени тома путь к его каталогу

wchar\_t\* path = buf\_path;

wcout << "Mount point: " << path << '\n';

if (GetDiskFreeSpaceEx(buf\_path, &free\_bytes, &total\_bytes, NULL) == 0) //получение значений количества свободных байт и общего обьема

cout << "Error: " << GetLastError() << '\n';

else

{

cout << "Free bytes : " << free\_bytes.QuadPart << '\n';

cout << "Total bytes: " << total\_bytes.QuadPart << '\n';

}

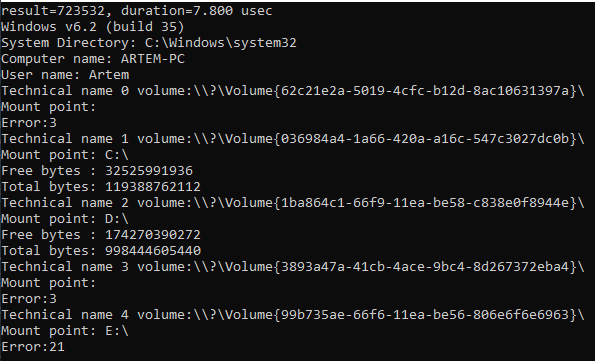
}while (FindNextVolume(search, buffer, sizeof(buffer))); //переход к следующему тому

if (GetLastError() != ERROR\_NO\_MORE\_FILES)

cout << "Error:" << GetLastError() << '\n'; //получение информации об ошибке

FindVolumeClose(search); //закрытие перебора томов

Результаты программы после запуска:



Сравнение с реальными данными:

