

МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ  
Государственное автономное профессиональное образовательное  
учреждение  
**«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»**  
**(ГАПОУ ОКЭИ)**

## **Практическая работа**

*ОКЭИ 09.02.07 7026 07 Ч*

Тема: «*Особенности использования функции VirtualProtect*»

Выполнил: Гадиляев Ислам Ильгамович

Оренбург 2026

## **Содержание**

1 Цель работы .....	3
2 Скриншот структуры готового проекта.....	3
3 Листинг программы .....	3
4 Результаты работы программы.....	5

Изм.	Лис	№ докум.	Подп.	Дата
Разраб.	Гадиляев И. И.			

ОКЭИ 09.02.07 7026 07 Ч

Практическая работа

Лит.	Лист	Листов
Ч	2	

Отделение программирования

## 1 Цель работы

Цель работы: получить практические навыки по использованию функции VirtualProtect и осуществить контроль за ошибками

Задание: модифицировать программу из ПЗ 9.

## 2 Скриншот структуры готового проекта

На рисунке 1 представлен скриншот структуры готового проекта.

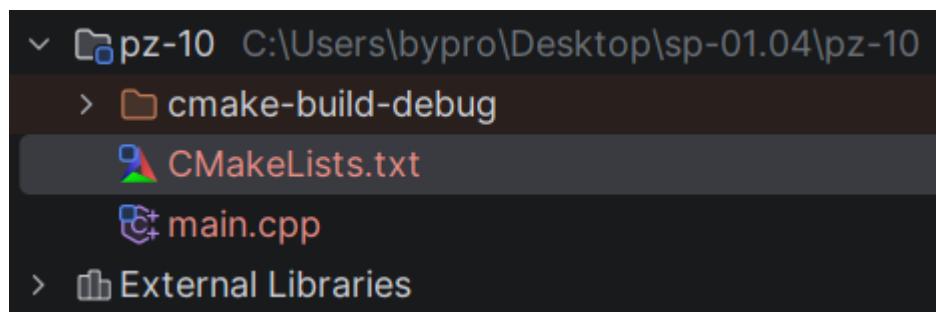


Рисунок 1 – Структура проекта

## 3 Листинг программы

Листинг файла main.cpp представлен ниже.

```
#include <iostream>
#include <windows.h>

using namespace std;

void PrintLastError(const char* msg)
{
    DWORD error = GetLastError();
    LPSTR buffer = nullptr;

    FormatMessageA(
        FORMAT_MESSAGE_ALLOCATE_BUFFER |
        FORMAT_MESSAGE_FROM_SYSTEM |
        FORMAT_MESSAGE_IGNORE_INSERTS,
        NULL,
        error,
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPSTR)&buffer,
        0,
        NULL
    );

    cout << msg << endl;
    cout << "Код ошибки: " << error << endl;
    if (buffer)
    {
        cout << "Описание: " << buffer << endl;
        LocalFree(buffer);
    }
}
```

Изм	Лист	№ докум.	Подп.	Дата	Лист
					3

```

        }

    int main()
    {
        SetConsoleOutputCP(CP_UTF8);

        SYSTEM_INFO systemInfo;
        GetSystemInfo(&systemInfo);
        DWORD pageSize = systemInfo.dwPageSize;

        cout << "Размер страницы: " << pageSize << " байт\n";

        LPVOID region = VirtualAlloc(
            NULL,
            pageSize,
            MEM_RESERVE,
            PAGE_READWRITE
        );

        if (!region)
        {
            PrintLastError("Ошибка MEM_RESERVE");
            return 1;
        }

        if (!VirtualAlloc(
            region,
            pageSize,
            MEM_COMMIT,
            PAGE_READWRITE))
        {
            PrintLastError("Ошибка MEM_COMMIT");
            return 1;
        }

        cout << "Регион памяти: " << region << endl;

        FillMemory(region, pageSize, 0x7F);
        cout << "Память заполнена значением 0x7F\n";

        cout << "Проверка чтения (до VirtualProtect): ";
        if (IsBadReadPtr(region, pageSize))
            cout << "недоступна\n";
        else
            cout << "доступна\n";

        DWORD oldProtect;
        if (!VirtualProtect(
            region,
            pageSize,
            PAGE_NOACCESS,
            &oldProtect))
        {
            PrintLastError("Ошибка VirtualProtect");
            return 1;
        }

        cout << "Атрибут защиты изменен на PAGE_NOACCESS\n";

        cout << "Проверка чтения (после VirtualProtect): ";
        if (IsBadReadPtr(region, pageSize))
            cout << "Недоступна\n";
        else
            cout << "доступна\n";

        MEMORY_BASIC_INFORMATION mbi;
        if (!VirtualQuery(region, &mbi, sizeof(mbi)))
        {

```

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

```

        PrintLastError("Ошибка VirtualQuery");
        return 1;
    }

    cout << "\nИнформация о регионе памяти:\n";
    cout << "BaseAddress: " << mbi.BaseAddress << endl;
    cout << "AllocationProtect: 0x" << hex << mbi.AllocationProtect << dec << endl;
    cout << "RegionSize:" << mbi.RegionSize << " байт\n";
    cout << "State: ";

    if (mbi.State == MEM_COMMIT)
        cout << "MEM_COMMIT\n";
    else if (mbi.State == MEM_RESERVE)
        cout << "MEM_RESERVE\n";
    else if (mbi.State == MEM_FREE)
        cout << "MEM_FREE\n";

    VirtualFree(region, 0, MEM_RELEASE);

    return 0;
}

```

## 4 Результаты работы программы

Результат выполнения программы показан на рисунке 2.

```

Run  pz_10 ×

C:\\bypro\Desktop\sp-01.04\pz-10\cmake-build-debug\pz_10.exe
Размер страницы: 4096 байт
Регион памяти: 0x16282e80000
Память заполнена значением 0x7F
Проверка чтения (до VirtualProtect): доступна
Атрибут защиты изменен на PAGE_NOACCESS
Проверка чтения (после VirtualProtect): Недоступна

Информация о регионе памяти:
BaseAddress: 0x16282e80000
AllocationProtect: 0x4
RegionSize: 4096 байт
State: MEM_COMMIT

Process finished with exit code 0

```

Рисунок 2 – Результат работы программы

Изм	Лист	№ докум.	Подп.	Дата	Лист
					ОКЭИ 09.02.07 7026 07 Ч 5