# Лабораторная работа: Получение дескриптора окна консоли, получение и изменение заголовка консоли

## **Цели:**

1. Изучить методы работы с консольным приложением в Windows.
2. Получить дескриптор окна консоли.
3. Изменить заголовок консоли.

## **Теоретический материал**

Консольные приложения в Windows позволяют взаимодействовать с пользователем через текстовый интерфейс. Одной из важных функций является управление заголовком окна консоли, что может улучшить пользовательский опыт. Заголовок консоли может содержать информацию о текущем состоянии программы или служить для ее идентификации.

### Основные функции работы с консолью:

* **GetConsoleWindow**: Получает дескриптор окна консоли.
* **SetConsoleTitle**: Устанавливает заголовок окна консоли.

## **Код программы:**

#include <windows.h>

#include <iostream>

#include <io.h>

#include <fcntl.h>

int main() {

// Установка кодировки консоли для корректного отображения русского текста

\_setmode(\_fileno(stdout), \_O\_U8TEXT);

\_setmode(\_fileno(stdin), \_O\_U8TEXT);

// Получение дескриптора окна консоли

HWND consoleWindow = GetConsoleWindow();

if (consoleWindow == NULL) {

std::wcerr << L"Ошибка: Не удалось получить дескриптор окна консоли." << std::endl;

return 1;

}

// Установка заголовка консоли

const wchar\_t\* newTitle = L"Мое Консольное Приложение"; // Устанавливаем новый заголовок

if (SetConsoleTitleW(newTitle)) {

std::wcout << L"Заголовок консоли успешно изменен на: " << newTitle << std::endl;

}

else {

std::wcerr << L"Ошибка: Не удалось изменить заголовок консоли." << std::endl;

return 1;

}

// Ожидание ввода для завершения программы

std::wcout << L"Нажмите Enter для выхода..." << std::endl;

std::wcin.get();

return 0;

}

### Пояснения к коду

1. **Установка кодировки**:
   * Используем функции \_setmode и \_fileno, чтобы установить режим кодировки для стандартного ввода и вывода на UTF-8. Это позволит корректно отображать русский текст.
2. **Получение дескриптора окна**:
   * Функция GetConsoleWindow возвращает дескриптор окна консоли. Если возвращается NULL, значит произошла ошибка.
3. **Установка заголовка консоли**:
   * Функция SetConsoleTitleW устанавливает новый заголовок окна консоли. Если заголовок успешно изменен, программа выводит соответствующее сообщение.
4. **Ожидание ввода**:
   * Используется std::wcin.get() для ожидания нажатия клавиши Enter перед закрытием консоли.

## **Результат выполнения**

После выполнения программы консольное окно должно отображать новый заголовок "Мое Консольное Приложение". Пользователь увидит сообщение об успешном изменении заголовка и будет ждать, пока нажмет клавишу Enter, чтобы выйти.

## **Самостоятельные задания**

1. **Измените заголовок консоли на свой текст**:
   * **Подсказка**: Измените строку const wchar\_t\* newTitle = L"Мое Консольное Приложение"; на ваш текст, например, L"Тестовое Приложение".
2. **Добавьте вывод текущего дескриптора окна**:
   * **Подсказка**: После получения дескриптора окна добавьте строку:

std::wcout << L"Дескриптор окна консоли: " << consoleWindow << std::endl;

1. **Проверьте наличие ошибок при изменении заголовка**:
   * **Подсказка**: Используйте GetLastError для получения кода ошибки, если функция SetConsoleTitleW возвращает FALSE. Выведите код ошибки в std::wcerr:

if (!SetConsoleTitleW(newTitle)) {

std::wcerr << L"Ошибка: " << GetLastError() << std::endl;

}

1. **Измените заголовок консоли на строку с переменной**:
   * **Подсказка**: Замените const wchar\_t\* newTitle на:

std::wstring title;

std::wcout << L"Введите новый заголовок: ";

std::getline(std::wcin, title);  
SetConsoleTitleW(title.c\_str());

1. **Создайте цикл для изменения заголовка**:
   * **Подсказка**: Введите цикл, который позволяет пользователю вводить новый заголовок до тех пор, пока он не введет "выход".

std::wstring title;

while (true) {

std::wcout << L"Введите новый заголовок (или 'выход' для завершения): ";

std::getline(std::wcin, title);

if (title == L"выход") break;

SetConsoleTitleW(title.c\_str());

}

## **Контрольные вопросы:**

1. Какой тип данных возвращает функция GetConsoleWindow?
2. Что произойдет, если вы попытаетесь установить заголовок консоли на пустую строку?
3. Как можно получить код ошибки, если операция не удалась?
4. Какой заголовок консоли был установлен после выполнения программы?
5. Можно ли изменить заголовок консоли на любой текст? Почему?
6. Какой тип аргумента принимает функция SetConsoleTitleW?
7. Что делает функция std::wcin.get() в конце программы?
8. Как можно получить информацию о состоянии консольного приложения через заголовок?
9. Какие функции Windows API связаны с работой с консолью?
10. Какой метод можно использовать для получения информации о текущем состоянии консоли?