Лабораторная работа: Определение максимального размера окна и установка координат окна консоли

Цель

Научиться определять максимальный размер окна консоли и изменять положение окна на экране. Освоить работу с Windows API для манипуляций с консольным окном.

Код программы

#include <windows.h>

#include <iostream>

#include <io.h>

#include <fcntl.h>

int main() {

// Установка кодировки для корректного отображения

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

// Получаем дескриптор окна консоли

HWND consoleWindow = GetConsoleWindow();

if (consoleWindow == NULL) {

std::wcout << L"Ошибка: не удалось получить дескриптор окна." << std::endl;

return 1;

}

// Определение максимального размера окна консоли

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD maxSize = GetLargestConsoleWindowSize(hConsole);

std::wcout << L"Максимальный размер окна в символах: "

<< maxSize.X << L" x " << maxSize.Y << std::endl;

std::wcout << L"Ожидание 3 секунды перед изменением размера окна..." << std::endl;

Sleep(3000);

// Установка новых размеров и координат

int newWidth = 800; // Ширина окна в пикселях

int newHeight = 600; // Высота окна в пикселях

int newPosX = 100; // Координата X

int newPosY = 100; // Координата Y

if (MoveWindow(consoleWindow, newPosX, newPosY, newWidth, newHeight, TRUE)) {

std::wcout << L"Размер и положение окна успешно изменены." << std::endl;

}

else {

std::wcout << L"Ошибка: не удалось изменить размер и положение окна." << std::endl;

}

// Завершение работы программы

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

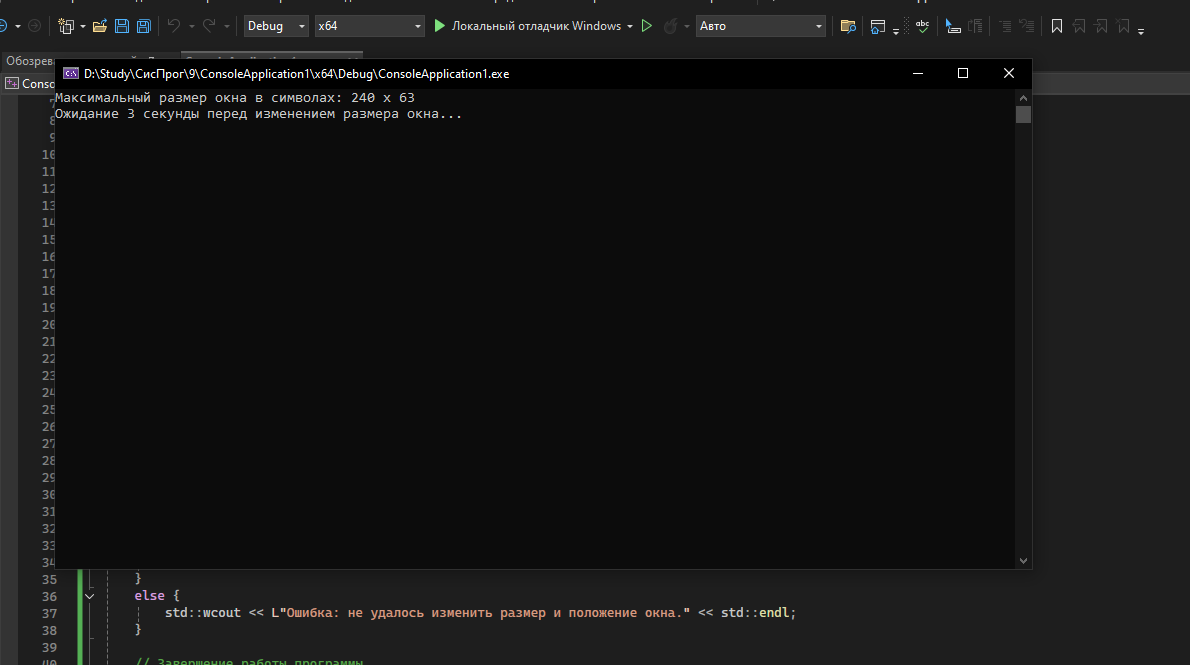
std::wcin.get();

return 0;

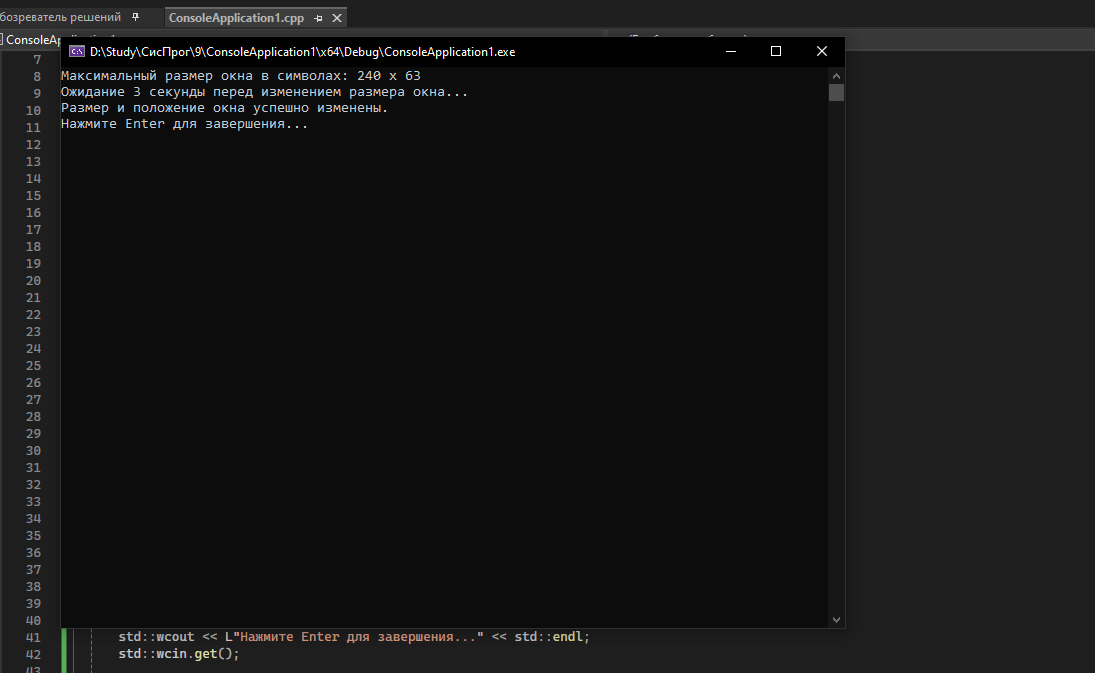
}

Результат:

До изменения:



После:



Самостоятельные задания

1. Измените положение окна на центр экрана.

Подсказка: Используйте разрешение экрана для вычисления координат центра. Например:

int screenWidth = GetSystemMetrics(SM\_CXSCREEN);

int screenHeight = GetSystemMetrics(SM\_CYSCREEN);

int newPosX = (screenWidth - newWidth) / 2;

int newPosY = (screenHeight - newHeight) / 2;

Код:

#include <windows.h>

#include <iostream>

#include <io.h>

#include <fcntl.h>

int main() {

// Установка кодировки для корректного отображения

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

// Получаем дескриптор окна консоли

HWND consoleWindow = GetConsoleWindow();

if (consoleWindow == NULL) {

std::wcout << L"Ошибка: не удалось получить дескриптор окна." << std::endl;

return 1;

}

// Определение максимального размера окна консоли

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD maxSize = GetLargestConsoleWindowSize(hConsole);

std::wcout << L"Максимальный размер окна в символах: "

<< maxSize.X << L" x " << maxSize.Y << std::endl;

// Установка новых размеров и координат

int newWidth; // Ширина окна в пикселях

int newHeight; // Высота окна в пикселях

//int newPosX = 100; // Координата X

//int newPosY = 100; // Координата Y

std::wcout << L"Введите новые ширину и длину"<< std::endl;

std::wcin >> newWidth >> newHeight;

int screenWidth = GetSystemMetrics(SM\_CXSCREEN);

int screenHeight = GetSystemMetrics(SM\_CYSCREEN);

int newPosX = (screenWidth - newWidth) / 2;

int newPosY = (screenHeight - newHeight) / 2;

std::wcout << L"Ожидание 3 секунды перед изменением размера окна..." << std::endl;

Sleep(3000);

if (MoveWindow(consoleWindow, newPosX, newPosY, newWidth, newHeight, TRUE)) {

std::wcout << L"Размер и положение окна успешно изменены." << std::endl;

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

GetConsoleScreenBufferInfo(hConsole, &csbi);

std::wcout << L"Текущий размер окна: " << csbi.dwSize.X << L" x " << csbi.dwSize.Y << std::endl;

}

else {

std::wcout << L"Ошибка: не удалось изменить размер и положение окна." << std::endl;

}

// Завершение работы программы

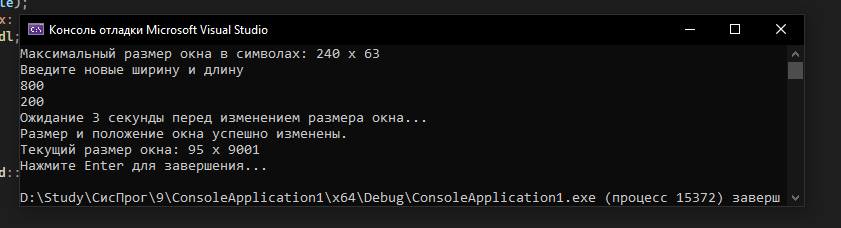
std::wcout << L"Нажмите Enter для завершения..." << std::endl;

std::wcin.get();

return 0;

}

Результат:



1. Проверьте текущий размер окна до его изменения.

Подсказка: Используйте функцию GetConsoleScreenBufferInfo перед MoveWindow для получения текущего размера окна.

Пример кода:

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

GetConsoleScreenBufferInfo(hConsole, &csbi);

std::wcout << L"Текущий размер окна: " << csbi.dwSize.X << L" x " << csbi.dwSize.Y << std::endl;

Код:

#include <windows.h>

#include <iostream>

#include <io.h>

#include <fcntl.h>

int main() {

// Установка кодировки для корректного отображения

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

// Получаем дескриптор окна консоли

HWND consoleWindow = GetConsoleWindow();

if (consoleWindow == NULL) {

std::wcout << L"Ошибка: не удалось получить дескриптор окна." << std::endl;

return 1;

}

// Определение максимального размера окна консоли

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD maxSize = GetLargestConsoleWindowSize(hConsole);

std::wcout << L"Максимальный размер окна в символах: "

<< maxSize.X << L" x " << maxSize.Y << std::endl;

std::wcout << L"Ожидание 3 секунды перед изменением размера окна..." << std::endl;

Sleep(3000);

// Установка новых размеров и координат

int newWidth = 800; // Ширина окна в пикселях

int newHeight = 600; // Высота окна в пикселях

int newPosX = 100; // Координата X

int newPosY = 100; // Координата Y

if (MoveWindow(consoleWindow, newPosX, newPosY, newWidth, newHeight, TRUE)) {

std::wcout << L"Размер и положение окна успешно изменены." << std::endl;

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

GetConsoleScreenBufferInfo(hConsole, &csbi);

std::wcout << L"Текущий размер окна: " << csbi.dwSize.X << L" x " << csbi.dwSize.Y << std::endl;

}

else {

std::wcout << L"Ошибка: не удалось изменить размер и положение окна." << std::endl;

}

// Завершение работы программы

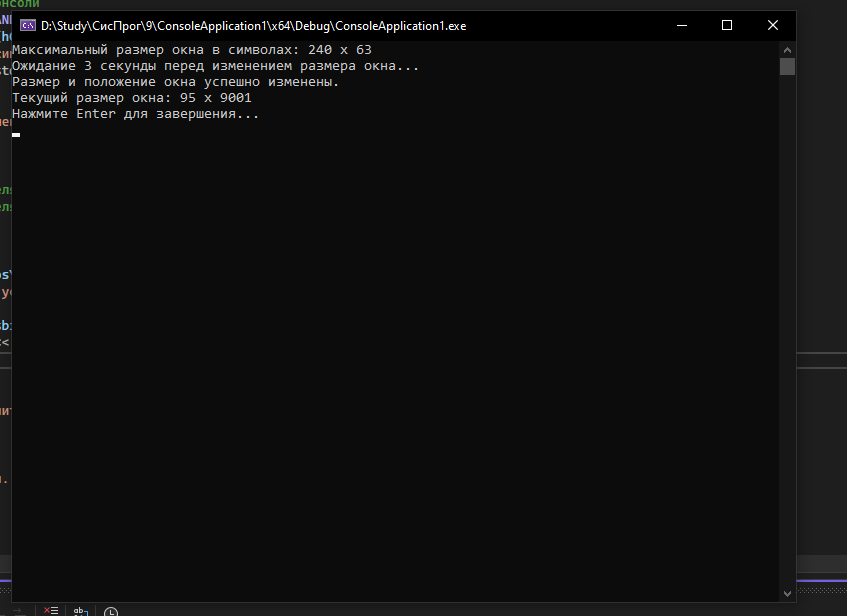
std::wcout << L"Нажмите Enter для завершения..." << std::endl;

std::wcin.get();

return 0;

}

Результат:



1. Добавьте возможность ввода размеров окна пользователем.

Подсказка: Используйте std::wcin для запроса новых значений newWidth и newHeight у пользователя.

Код:

#include <windows.h>

#include <iostream>

#include <io.h>

#include <fcntl.h>

int main() {

// Установка кодировки для корректного отображения

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

// Получаем дескриптор окна консоли

HWND consoleWindow = GetConsoleWindow();

if (consoleWindow == NULL) {

std::wcout << L"Ошибка: не удалось получить дескриптор окна." << std::endl;

return 1;

}

// Определение максимального размера окна консоли

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD maxSize = GetLargestConsoleWindowSize(hConsole);

std::wcout << L"Максимальный размер окна в символах: "

<< maxSize.X << L" x " << maxSize.Y << std::endl;

// Установка новых размеров и координат

int newWidth; // Ширина окна в пикселях

int newHeight; // Высота окна в пикселях

int newPosX = 100; // Координата X

int newPosY = 100; // Координата Y

std::wcout << L"Введите новые ширину и длину"<< std::endl;

std::wcin >> newWidth >> newHeight;

std::wcout << L"Ожидание 3 секунды перед изменением размера окна..." << std::endl;

Sleep(3000);

if (MoveWindow(consoleWindow, newPosX, newPosY, newWidth, newHeight, TRUE)) {

std::wcout << L"Размер и положение окна успешно изменены." << std::endl;

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

GetConsoleScreenBufferInfo(hConsole, &csbi);

std::wcout << L"Текущий размер окна: " << csbi.dwSize.X << L" x " << csbi.dwSize.Y << std::endl;

}

else {

std::wcout << L"Ошибка: не удалось изменить размер и положение окна." << std::endl;

}

// Завершение работы программы

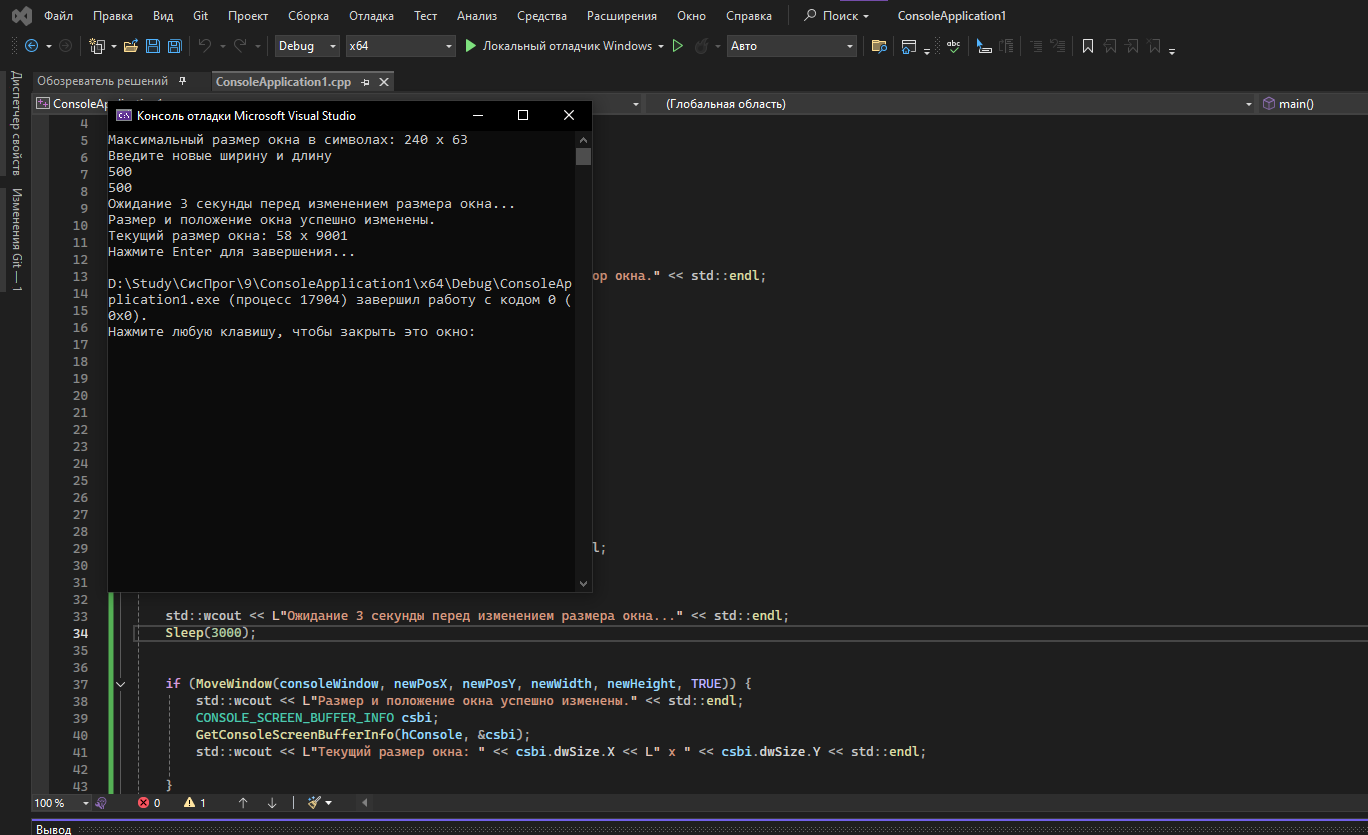
std::wcout << L"Нажмите Enter для завершения..." << std::endl;

std::wcin.get();

return 0;

}

Результат:



1. Выведите позицию окна после его перемещения.

Подсказка: Используйте GetWindowRect после вызова MoveWindow и выведите координаты.

Код:

#include <windows.h>

#include <iostream>

#include <io.h>

#include <fcntl.h>

int main() {

// Установка кодировки для корректного отображения

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

// Получаем дескриптор окна консоли

HWND consoleWindow = GetConsoleWindow();

if (consoleWindow == NULL) {

std::wcout << L"Ошибка: не удалось получить дескриптор окна." << std::endl;

return 1;

}

// Определение максимального размера окна консоли

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD maxSize = GetLargestConsoleWindowSize(hConsole);

std::wcout << L"Максимальный размер окна в символах: "

<< maxSize.X << L" x " << maxSize.Y << std::endl;

// Установка новых размеров и координат

int newWidth; // Ширина окна в пикселях

int newHeight; // Высота окна в пикселях

//int newPosX = 100; // Координата X

//int newPosY = 100; // Координата Y

std::wcout << L"Введите новые ширину и длину" << std::endl;

std::wcin >> newWidth >> newHeight;

int screenWidth = GetSystemMetrics(SM\_CXSCREEN);

int screenHeight = GetSystemMetrics(SM\_CYSCREEN);

int newPosX = (screenWidth - newWidth) / 2;

int newPosY = (screenHeight - newHeight) / 2;

std::wcout << L"Ожидание 3 секунды перед изменением размера окна..." << std::endl;

Sleep(3000);

if (MoveWindow(consoleWindow, newPosX, newPosY, newWidth, newHeight, TRUE)) {

std::wcout << L"Размер и положение окна успешно изменены." << std::endl;

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

GetConsoleScreenBufferInfo(hConsole, &csbi);

std::wcout << L"Текущий размер окна: " << csbi.dwSize.X << L" x " << csbi.dwSize.Y << std::endl;

}

else {

std::wcout << L"Ошибка: не удалось изменить размер и положение окна." << std::endl;

}

// Завершение работы программы

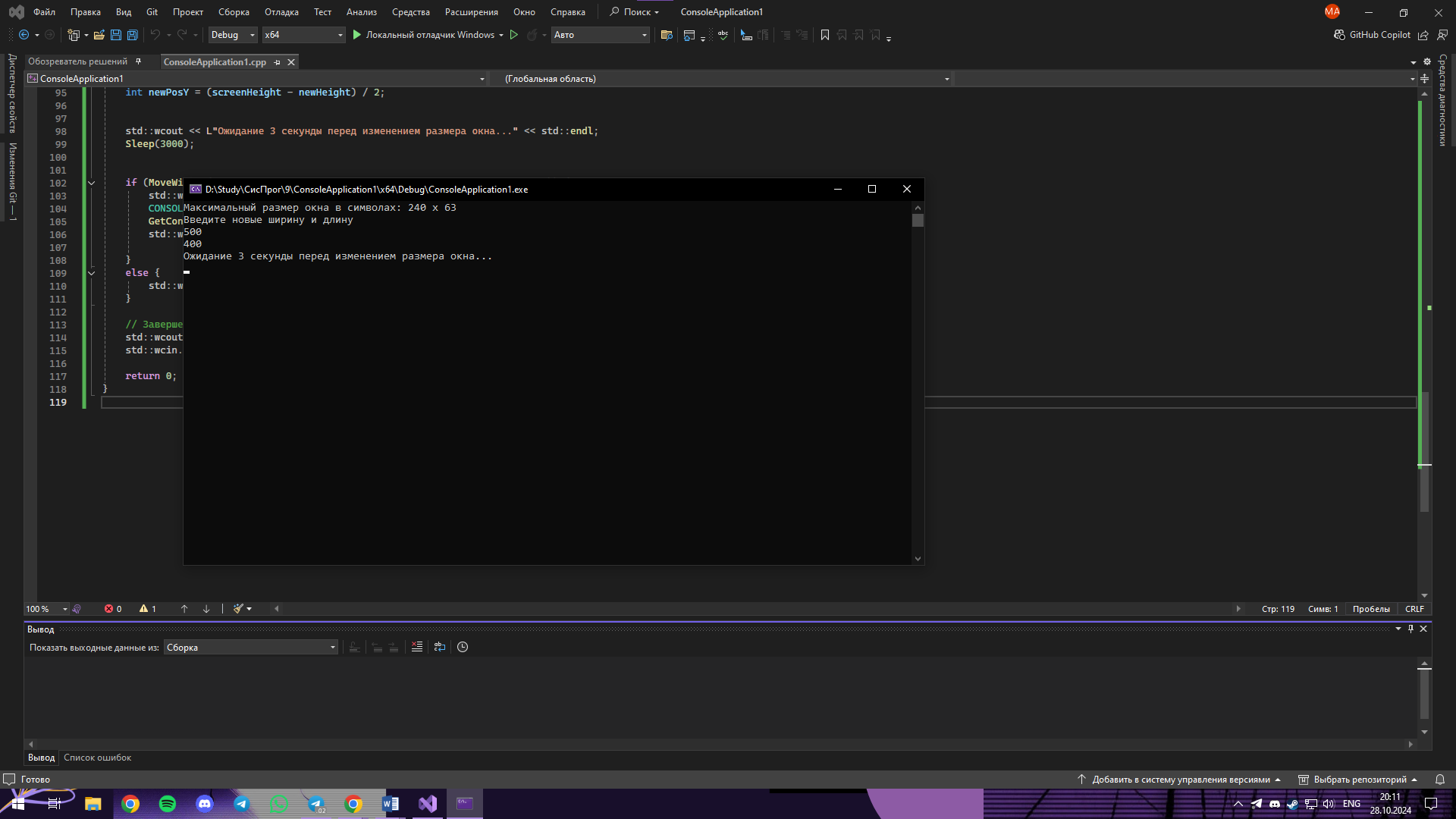
std::wcout << L"Нажмите Enter для завершения..." << std::endl;

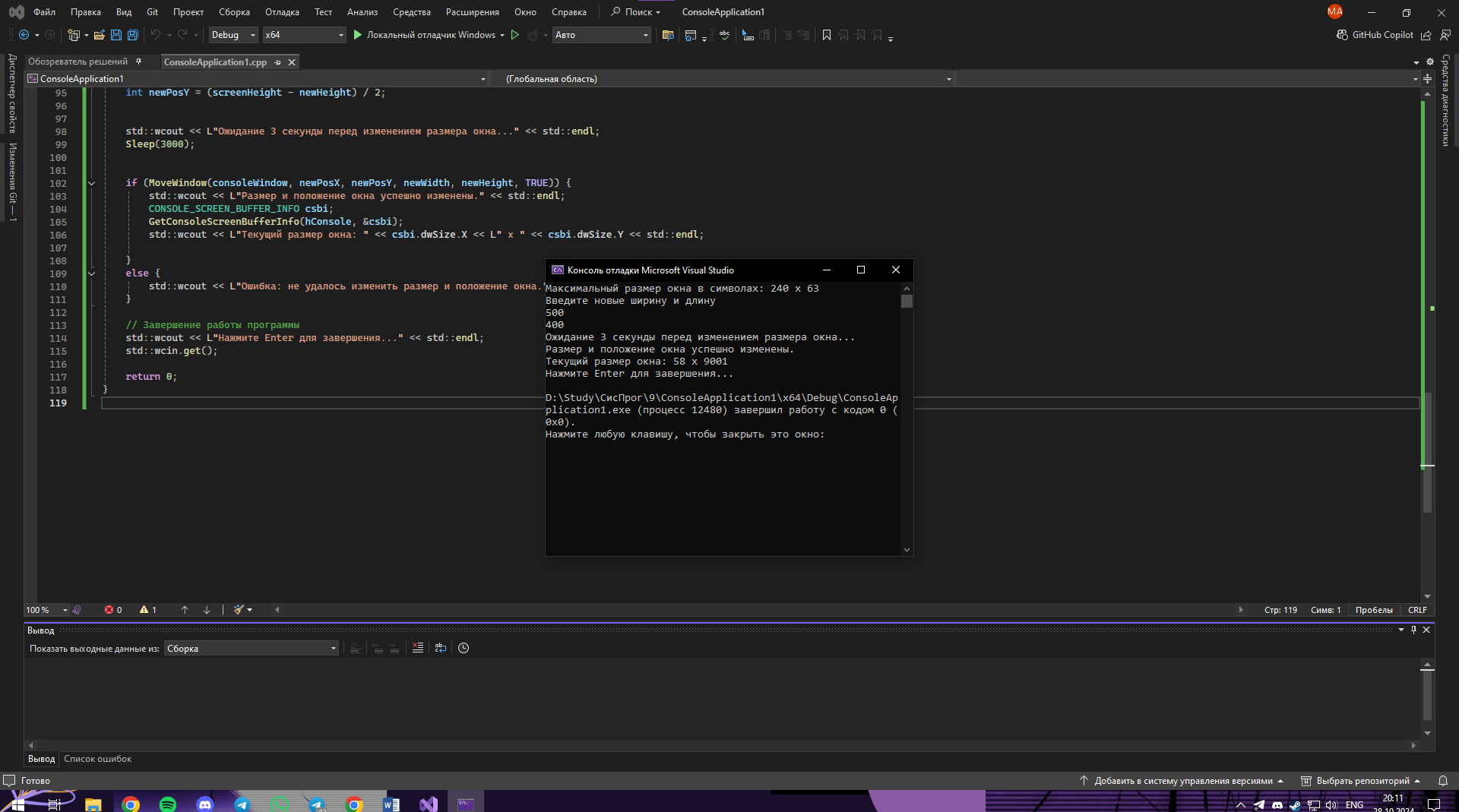
std::wcin.get();

return 0;

}

Результат:





Контрольные вопросы

1. Что возвращает функция GetLargestConsoleWindowSize?

Функция GetLargestConsoleWindowSize возвращает максимальный размер окна консоли в символах (ширина и высота) для текущего шрифта консоли и экрана.

1. Как узнать текущий размер и позицию консольного окна?

Для получения текущего размера и позиции консольного окна используется функция GetConsoleScreenBufferInfo, которая возвращает структуру CONSOLE\_SCREEN\_BUFFER\_INFO, содержащую информацию о размере окна и его текущих координатах.

1. Как работает функция MoveWindow?

Функция MoveWindow изменяет положение и размер указанного окна на экране. Она принимает дескриптор окна, новые координаты (X и Y), новые размеры (ширина и высота) и логическое значение, указывающее, нужно ли обновить окно немедленно.

1. Что будет, если задать размеры окна, превышающие максимальные значения?

Если задать размеры окна, превышающие максимальные значения, функция MoveWindow может вернуть ошибку, и окно не будет изменено. Размеры окна будут ограничены максимальными значениями, установленными для консоли.

1. Какие параметры нужны для изменения положения окна на экране?

Для изменения положения окна на экране нужны координаты X и Y, которые определяют новое местоположение верхнего левого угла окна.

1. Почему важно использовать дескриптор окна?

Дескриптор окна (HWND) является уникальным идентификатором для каждого окна в Windows. Он необходим для выполнения операций с конкретным окном, таких как изменение его размера, положения или состояния.

1. Как можно изменить размеры консоли в символах, а не в пикселях?

Для изменения размеров консоли в символах можно использовать функции SetConsoleScreenBufferSize и SetConsoleWindowInfo, которые работают с размерами в символах, а не в пикселях.

1. Какой код нужно использовать для получения разрешения экрана?

Для получения разрешения экрана можно использовать функцию GetSystemMetrics с параметрами SM\_CXSCREEN (ширина) и SM\_CYSCREEN (высота):

int screenWidth = GetSystemMetrics(SM\_CXSCREEN);

int screenHeight = GetSystemMetrics(SM\_CYSCREEN);

1. Какие значения возвращаются для SM\_CXSCREEN и SM\_CYSCREEN?

SM\_CXSCREEN возвращает ширину экрана в пикселях, а SM\_CYSCREEN возвращает высоту экрана в пикселях.

1. В чем разница между GetLargestConsoleWindowSize и GetConsoleScreenBufferInfo?

GetLargestConsoleWindowSize возвращает максимальный размер окна консоли в символах, в то время как GetConsoleScreenBufferInfo возвращает информацию о текущем размере и положении буфера экрана, включая текущие размеры окна, которые могут быть меньше максимальных значений.