Для создания клиентской программы, которая будет отправлять сообщения серверу, можно использовать API Windows Sockets (Winsock). Клиентская программа будет подключаться к серверу, отправлять сообщения и получать ответы.

Вот пример клиентской программы на C++, которая подключается к серверу, отправляет сообщения и получает ответы:

\*\*Client.cpp\*\*:

```cpp

#include <iostream>

#include <winsock2.h>

#include <ws2tcpip.h>

#pragma comment(lib, "Ws2\_32.lib")

int main() {

WSADATA wsaData;

SOCKET clientSocket;

sockaddr\_in serverAddr;

char buffer[512];

int result;

// Инициализация Winsock

result = WSAStartup(MAKEWORD(2, 2), &wsaData);

if (result != 0) {

std::cerr << "WSAStartup failed: " << result << std::endl;

return 1;

}

// Создание сокета

clientSocket = socket(AF\_INET, SOCK\_STREAM, 0);

if (clientSocket == INVALID\_SOCKET) {

std::cerr << "Socket creation failed: " << WSAGetLastError() << std::endl;

WSACleanup();

return 1;

}

// Настройка адреса сервера

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(3487); // Порт сервера

inet\_pton(AF\_INET, "127.0.0.1", &serverAddr.sin\_addr); // IP-адрес сервера

// Подключение к серверу

result = connect(clientSocket, (sockaddr\*)&serverAddr, sizeof(serverAddr));

if (result == SOCKET\_ERROR) {

std::cerr << "Connection failed: " << WSAGetLastError() << std::endl;

closesocket(clientSocket);

WSACleanup();

return 1;

}

std::cout << "Connected to server" << std::endl;

// Отправка сообщений серверу

std::string message;

while (true) {

std::cout << "Enter message: ";

std::getline(std::cin, message);

if (message == "exit") {

break;

}

result = send(clientSocket, message.c\_str(), message.length(), 0);

if (result == SOCKET\_ERROR) {

std::cerr << "Send failed: " << WSAGetLastError() << std::endl;

break;

}

result = recv(clientSocket, buffer, sizeof(buffer), 0);

if (result > 0) {

std::cout << "Received from server: " << std::string(buffer, result) << std::endl;

} else if (result == 0) {

std::cout << "Connection closed by server" << std::endl;

break;

} else {

std::cerr << "Recv failed: " << WSAGetLastError() << std::endl;

break;

}

}

// Закрытие сокета и очистка Winsock

closesocket(clientSocket);

WSACleanup();

return 0;

}

```

### Пошаговое объяснение

1. \*\*Инициализация Winsock\*\*:

```cpp

result = WSAStartup(MAKEWORD(2, 2), &wsaData);

if (result != 0) {

std::cerr << "WSAStartup failed: " << result << std::endl;

return 1;

}

```

2. \*\*Создание сокета\*\*:

```cpp

clientSocket = socket(AF\_INET, SOCK\_STREAM, 0);

if (clientSocket == INVALID\_SOCKET) {

std::cerr << "Socket creation failed: " << WSAGetLastError() << std::endl;

WSACleanup();

return 1;

}

```

3. \*\*Настройка адреса сервера\*\*:

```cpp

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(3487); // Порт сервера

inet\_pton(AF\_INET, "127.0.0.1", &serverAddr.sin\_addr); // IP-адрес сервера

```

4. \*\*Подключение к серверу\*\*:

```cpp

result = connect(clientSocket, (sockaddr\*)&serverAddr, sizeof(serverAddr));

if (result == SOCKET\_ERROR) {

std::cerr << "Connection failed: " << WSAGetLastError() << std::endl;

closesocket(clientSocket);

WSACleanup();

return 1;

}

```

5. \*\*Отправка сообщений серверу и получение ответов\*\*:

```cpp

std::string message;

while (true) {

std::cout << "Enter message: ";

std::getline(std::cin, message);

if (message == "exit") {

break;

}

result = send(clientSocket, message.c\_str(), message.length(), 0);

if (result == SOCKET\_ERROR) {

std::cerr << "Send failed: " << WSAGetLastError() << std::endl;

break;

}

result = recv(clientSocket, buffer, sizeof(buffer), 0);

if (result > 0) {

std::cout << "Received from server: " << std::string(buffer, result) << std::endl;

} else if (result == 0) {

std::cout << "Connection closed by server" << std::endl;

break;

} else {

std::cerr << "Recv failed: " << WSAGetLastError() << std::endl;

break;

}

}

```

6. \*\*Закрытие сокета и очистка Winsock\*\*:

```cpp

closesocket(clientSocket);

WSACleanup();

```

Этот пример демонстрирует, как создать клиентскую программу, которая подключается к серверу, отправляет сообщения и получает ответы. Клиентская программа использует функции Winsock для создания сокета, подключения к серверу, отправки и получения данных.