

Лабораторная работа 1

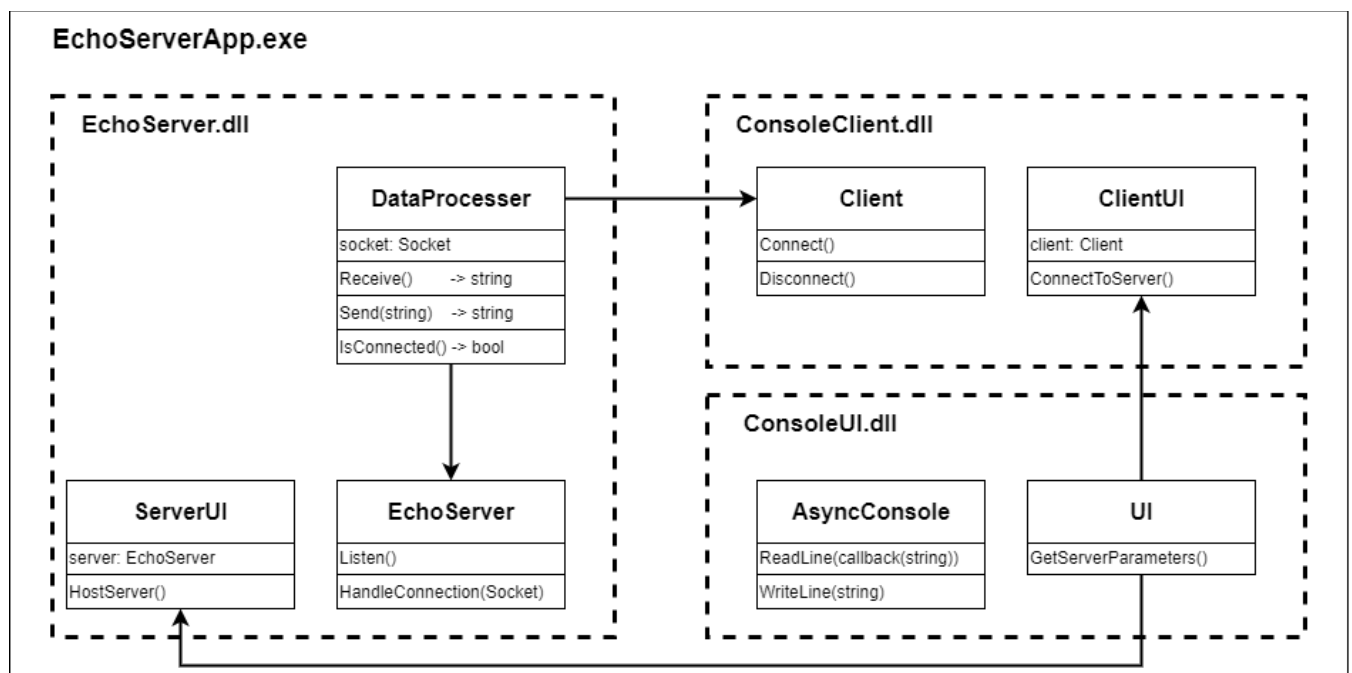
«Эхо сервер»

Целью этой лабораторной работы является разработка клиент-серверного приложения на языке C#. Клиент должен отсылать на сервер ввод пользователя, а сервер - посылать в ответ полученные данные.

Для сетевого взаимодействия стандартная библиотека .NET Core предоставляет класс `Socket`, с помощью экземпляров которого можно устанавливать соединения и передавать по ним данные с использованием одного из предложенных протоколов.

Организация проекта

Проект компилируется в один исполняемый файл, совмещающий в себе функционал как сервера, так и клиента, при запуске пользователю предлагается выбрать режим в котором будет работать приложение (сервер или клиент). Реализации же серверной и клиентской вынесены в две разные библиотеки: `EchoServer.dll` и `ConsoleClient.dll`

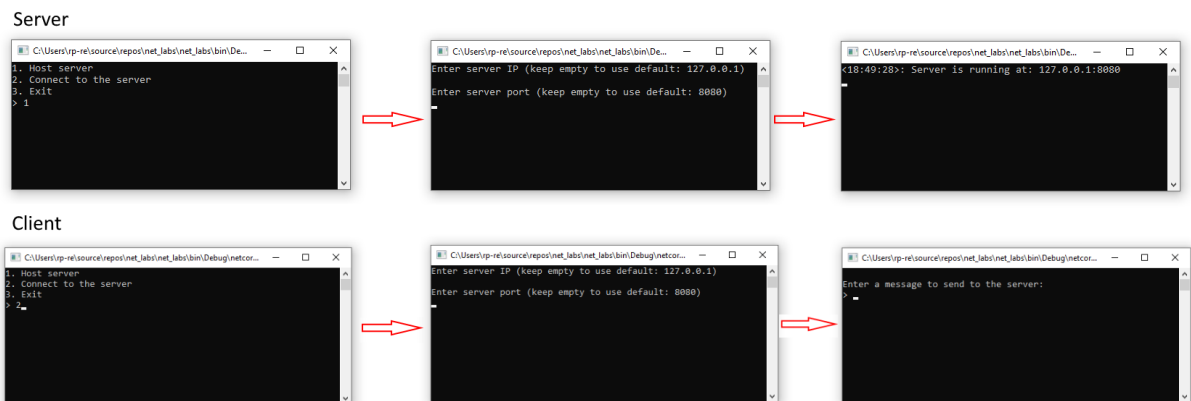


Общая для клиента и сервера логика по сетевому взаимодействию: получение и отправка текстовых сообщений, вынесена в их общий базовый класс `DataProcessor`. Помимо непосредственно сервера и клиента, библиотеки содержат классы `ServerUI` и `ClientUI`, которые предоставляют консольные интерфейсы взаимодействия с основными классами.

Класс AsyncConsole реализует возможность одновременного вывода и чтения с консоли, этот класс будет использоваться в следующих лабораторных.

Использование приложения

При запуске пользователю предлагается выбрать режим работы приложения и ввести параметры сервера:



После чего, в случае сервера: создается экземпляр класса EchoServer с заданными параметрами, у которого вызывается метод Listen, представляющий собой цикл, бесконечно принимающий новые соединения.

```
public void Listen() {  
    socket.Listen();  
    while (true) {  
        Socket client = socket.Accept();  
        HandleConnection(client);  
    }  
}
```

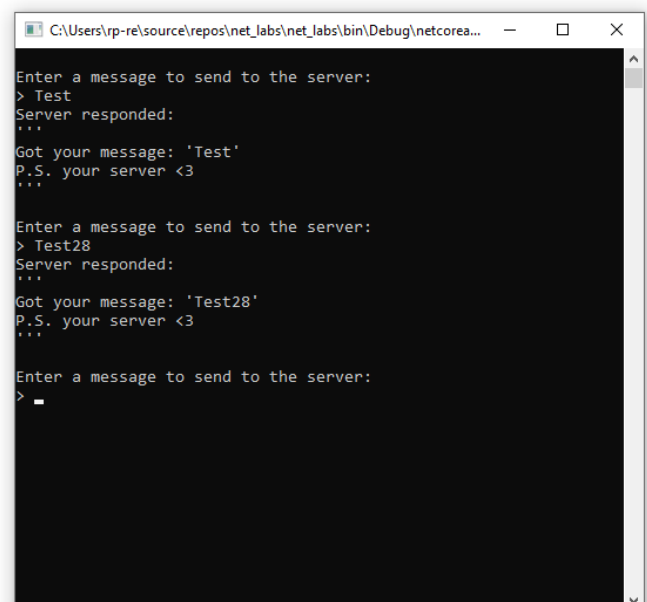
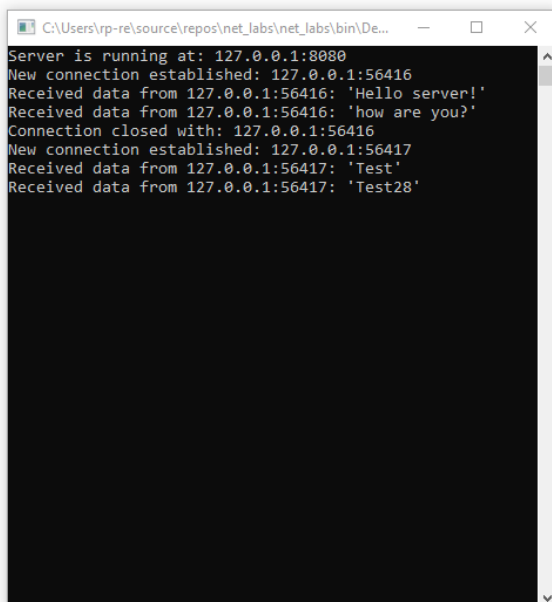
Как только соединение установлено, новый сокет, олицетворяющий это соединение, передается в HandleConnection, принимающий сообщения от клиента вплоть до потери связи с ним:

```
public virtual void HandleConnection(Socket client) {  
    while (true) {  
        if (!client.IsConnected()) break;  
        string request = Receive(client);  
        string response = ProcessMessage(request);  
        Send(response, client);  
    }  
    client.Shutdown(SocketShutdown.Both);  
    client.Close();  
}
```

Похожая ситуация и в случае с клиентом, он в бесконечном цикле отправляет набранный текст и получает ответы от сервера до тех пор, пока соединение не разорвется или пока пользователь не нажмет CTRL+Z - в этом случае клиент инициирует отсоединение от сервера:

```
while (true) {  
    if (!client.IsConnected()) break;  
    string request = Console.ReadLine();  
    if (request == CTRL_Z) break;  
    string response = client.SendReceive(request);  
    Console.WriteLine(response);  
}  
client.Disconnect();
```

Пример работы программы:



Приложения

1. Код программы на GitHub: <https://github.com/proxodilka/web-labs>