LABORATORIUM SIECI KOMPUTEROWYCH

|  |  |
| --- | --- |
| Data wykonania ćwiczenia: | **19.05.2020** |
| Rok studiów: | **2** |
| Semestr: | 4 |
| Grupa studencka: | **1** |
| Grupa laboratoryjna: | **B** |

|  |
| --- |
| *9* |

# Ćwiczenie nr

Temat: **Aplikacje klient-serwer.**

Osoby wykonujące ćwiczenia:

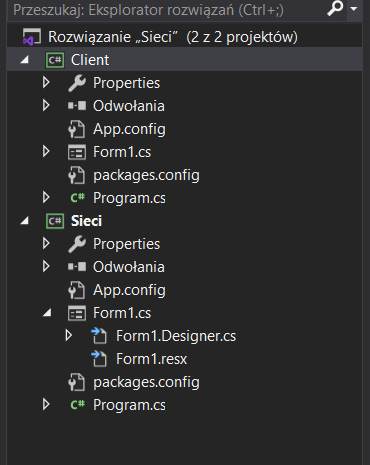
*1. Marek Żyła*

*2. Michał Krzyżowski*

### Katedra Informatyki i Automatyki

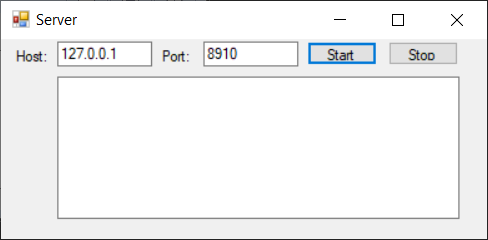
Na obecnych laboratoriach należało wykonać aplikację klient-server. Aplikacja miała być wykonana w dowolnym języku programowania – my napisaliśmy ją w c#. Celem aplikacji była komunikacja między dwoma użytkownikami, gdzie dodatkowym wymogiem było by aplikacja składała się z 2 programów – żądającego usług i tego który przetwarza.

W finalnej postaci nasz program ma poniższą budowę:

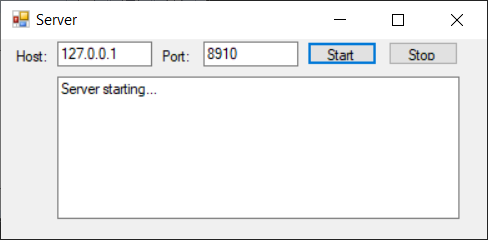


Widzimy tu dwa programy odpowiadające za Server (sieci) i Client (klient).

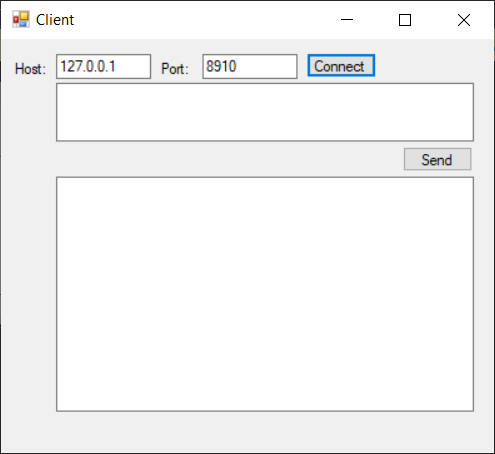
Poniżej prezentujemy okno Servera:



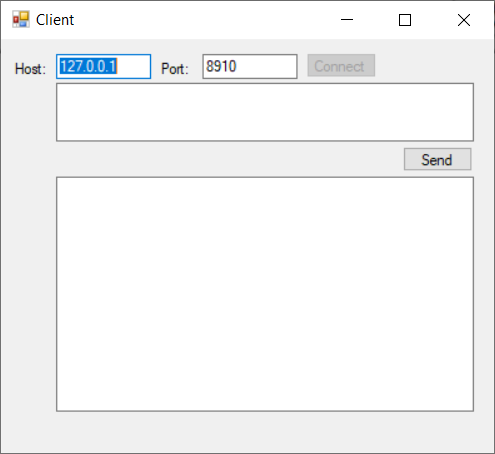
Możemy w tym oknie rozpocząć pracę serwera poprzez naciśnięcie przycisku start:



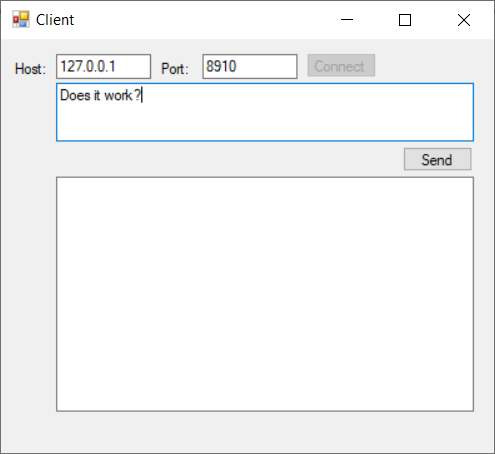
W tym czasie w osobnym oknie odpalamy nasz drugi program – klient:



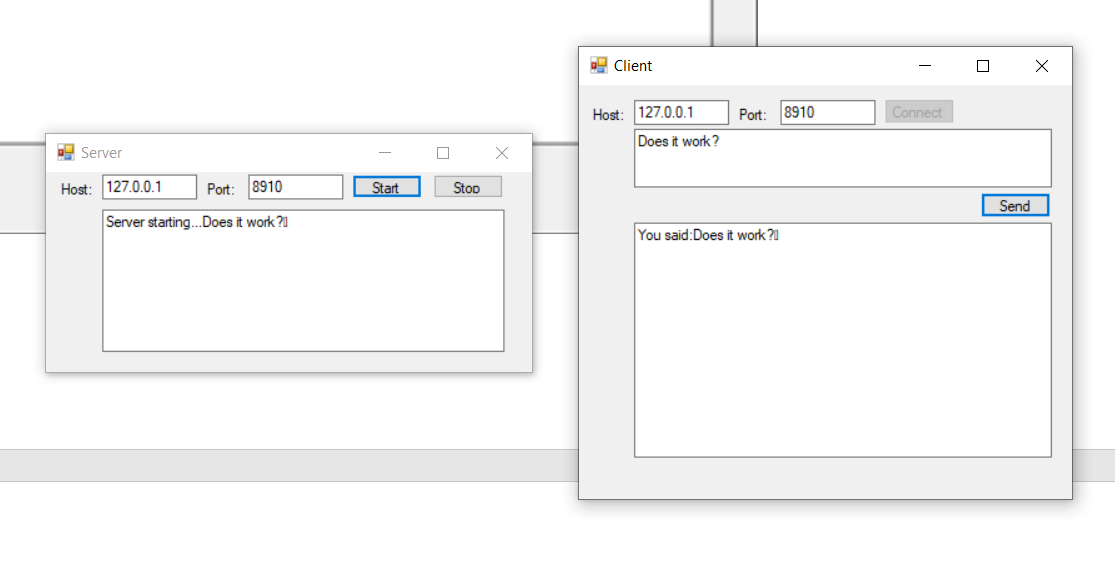
Klikamy przycisk połącz – czego efektem jest zablokowanie dostępu do przycisku.



Następnie wpisujemy coś w oknie poprzedzającym przycisk Send i naciskamy na przycisk:

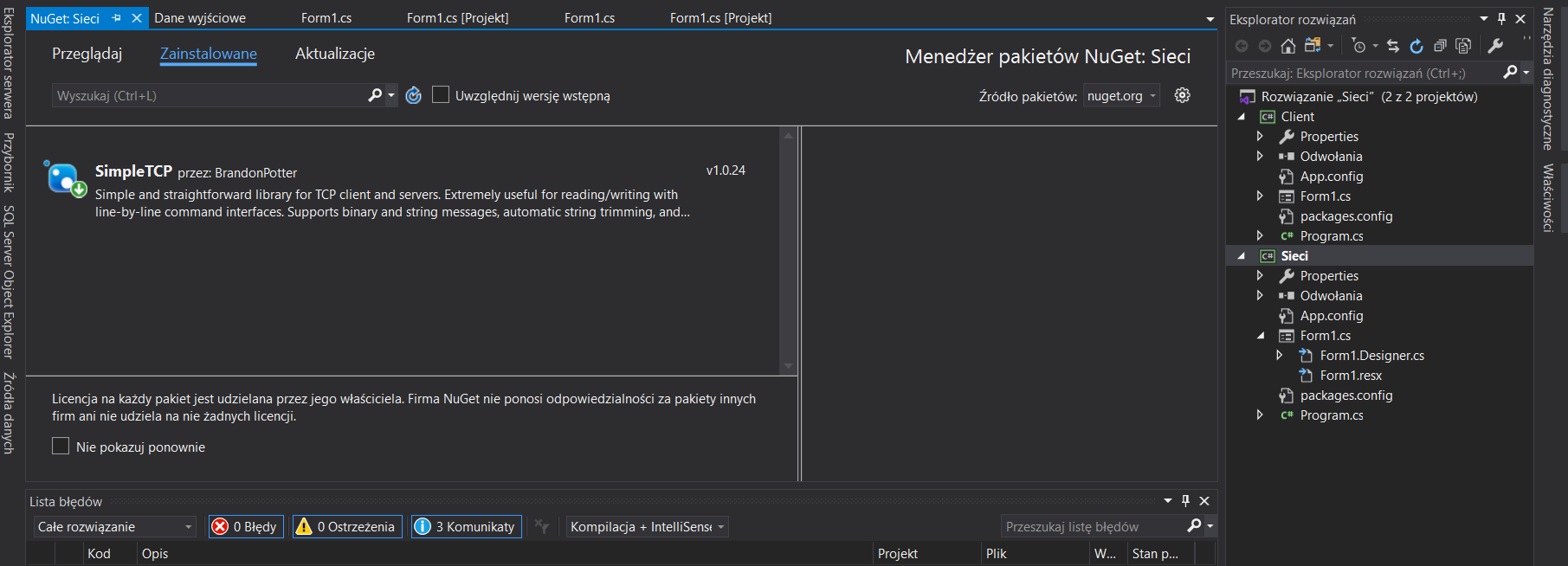


Efekt:

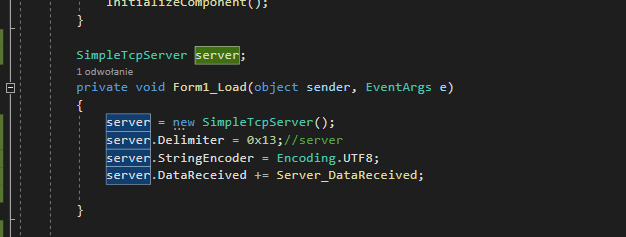


Jak widać komunikacja między serwerem i naszym klientem przebiega sprawnie.

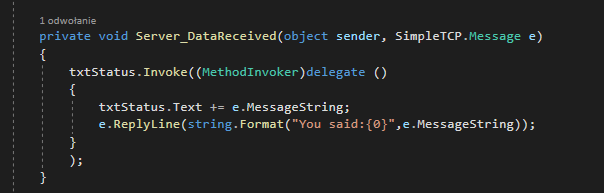
Teraz jeżeli idzie o kod w programie, całość działa dzięki doinstalowanemu pakietkowi NuGet: „SimpleTCP” – udostępnia on nam opcje połączeniowe:



Na samym początku przy wczytywaniu naszego formularzu tworzymy zmienną typu SimpleTcpServer zmienna ta będzie reprezentowała nasz server (wprowadzamy jej podstawowe informacje jak np. kodowanie ustawione na utf-8):

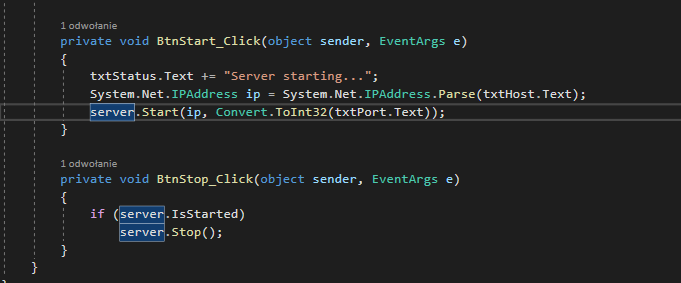


Dopisaliśmy też event o nazwie Server\_DataReceived:

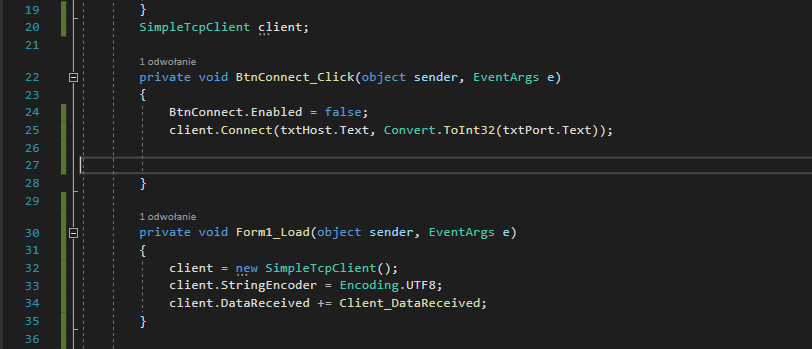


Odpowiada on za poprawne doświetlanie informacji w textboxie.

Poniżej mamy jeszcze obsługę dwóch naszych przycisków na serwerze jeden odpowiada za rozpoczęcie pracy, a drugi za zakończenie:

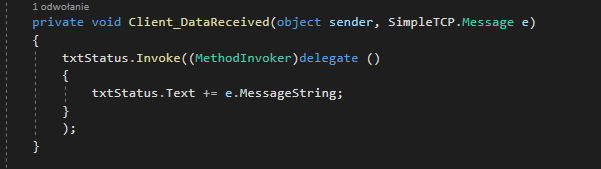


Jeżeli idzie o klient jest on zbudowany analogicznie do Serwera:



Zmienia się nasz typ którym jest SimpleTcpClient i pojawia się metoda Connect odpowiadająca za połączenie - aktywowana po naciśnięciu przycisku connect.

Jest też dopisany „wyłapywacz” danych podobnie jak na serwerze:



No i dodatkowo okodowane działanie przycisku „send” z ustawiony opóźnieniem 3-sek w wyświetlaniu.

