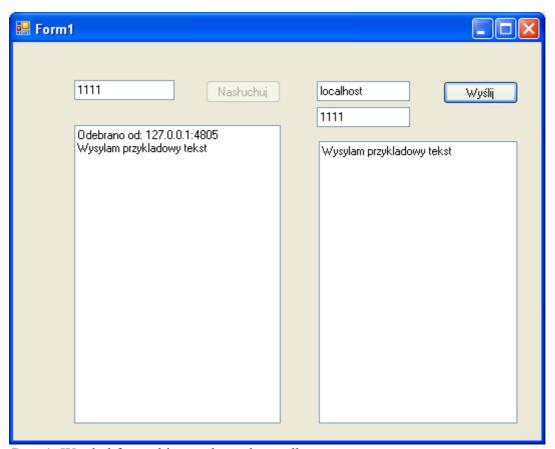
Tworzenie programów dla platformy .NET Zajęcia laboratoryjne

Ćwiczenie nr 4 Komponenty i kontrolki

W ramach ćwiczenia należy napisać prosty komunikator sieciowy, który można osadzać w innych aplikacjach.

Aplikacja składa się z następujących elementów:

- 1. **Komponentu** komunikacyjnego, który:
 - a. Całkowicie abstrahuje od GUI;
 - b. Wykorzystuje gniazdka System.Net, System.Net.Sockets;
 - c. Udostępnia zdarzenie OdebranoEvent, wywoływane po odebraniu wiadomości od innego komunikatora, gdzie przekazanym argumentem jest obiekt klasy, przechowujący nadawcę oraz treść komunikatu delegate, event, dziedziczenie z EventArgs;
 - d. Udostępnia metodę Nasluchuj (int port), która:
 - i. Tworzy, podpina i nasłuchuje na wskazanym porcie Socket new, Bind, Listen;
 - ii. Tworzy i uruchamia wątek roboczy **System.Threading.Thread, Thread.Start**, który:
 - iii. Oczekuje na przychodzące połączenia **Socket.Accept**, a następnie odbiera od nich komunikaty **Socket.Receive**;
 - iv. W przypadku odbioru komunikatu, wywoływane jest zdarzenie OdebranoEvent.
 - v. Należy zadbać, by po zamknięciu formatki wątek roboczy kończył działanie **Thread.Abort**.
 - e. Udostępnia metodę Wyslij (string adres, int port, string tresc), która:
 - i. Łączy się z gniazdkiem nasłuchującym- Socket.Connect;
 - ii. Wysyła treść komunikatu Socket.Send;
 - iii. Rozłącza się Socket.Close;
- 2. **Kontrolki**, która:
 - a. Wykorzystuje powyższy komponent komunikacyjny dodać referencję i using;
 - b. Udostępnia interfejs umożliwiający wysyłanie i odbieranie komunikatów (rys. 1);
 - c. Dostarcza procedurę obsługi zdarzenia OdebranoEvent. Ważne! Aby uniknąć konfliktu kontekstów wątków, wpisywanie tekstu do kontrolek należy wykonywać poprzez Form.Invoke;
- 3. Formatki, która:
 - a. Zawiera jedynie osadzoną kontrolkę.



Rys. 1 Wygląd formatki z osadzoną kontrolką