

Object.cs

```
using System;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;

namespace RemotingSamples {

    public class HelloServer : MarshalByRefObject {

        public HelloServer() {
            Console.WriteLine("Utworzono instancję HelloServer");
        }

        public String HelloMethod(String name) {
            Console.WriteLine("Hello.HelloMethod : {0}", name);
            return "Hi there " + name;
        }

    }

}
```

Kompilacja zdalnego obiektu:

```
csc /debug+ /target:library /out:object.dll object.cs
```

Server.cs

```
using System;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;

namespace RemotingSample {
    public class Sample {
        public static int Main(string [] args) {

// Otwarcie nowego kanalu HTTP lub TCP i zarejestrowanie go

//     HttpChannel channel = new HttpChannel(1025);
//     TcpChannel channel = new TcpChannel(8085);
//     ChannelServices.RegisterChannel(channel);

// Zarejestrowanie obiektu RemotingSample.HelloServer jako SingleCall lub Singleton

        RemotingConfiguration.RegisterWellKnownServiceType(
            Type.GetType("RemotingSamples.HelloServer,object"),
            "SayHello",
            WellKnownObjectMode.SingleCall);

//     RemotingConfiguration.RegisterWellKnownServiceType(
//         Type.GetType("RemotingSamples.HelloServer,object"),
//         "SayHello",
//         WellKnownObjectMode.Singleton);

        System.Console.WriteLine("Nacisnij aby zakonczyc prace serwera...");
        System.Console.ReadLine(); return 0;
    }
}

csc /debug+ /r:object.dll /r:System.Runtime.Remoting.dll server.cs
```

Client.cs

```
using System;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;

namespace RemotingSamples {

    public class Client
    {
        public static int Main(string [] args)
        {
            // Otwarcie i zarejestrowanie odpowiedniego kanału (analogicznie jak w serwerze!)

            //   HttpChannel channel = new HttpChannel();
            //   TcpChannel channel = new TcpChannel();
            //   ChannelServices.RegisterChannel(channel);

            // Pobranie referencji do zdalnego obiektu (tu localhost)

            //   HelloServer obj = (HelloServer)Activator.GetObject(
            //                                   typeof(RemotingSamples.HelloServer),
            //                                   "http://localhost:1025/SayHello");

            HelloServer obj = (HelloServer)Activator.GetObject(
                                typeof(RemotingSamples.HelloServer),
                                "tcp://localhost:8085/SayHello");

            if (obj == null)
                Console.WriteLine("Could not locate server");
            else
                Console.WriteLine(obj.HelloMethod("Caveman")); return 0;
        }
    }
}

csc /debug+ /r:object.dll /r:server.exe /r:System.Runtime.Remoting.dll client.cs
```