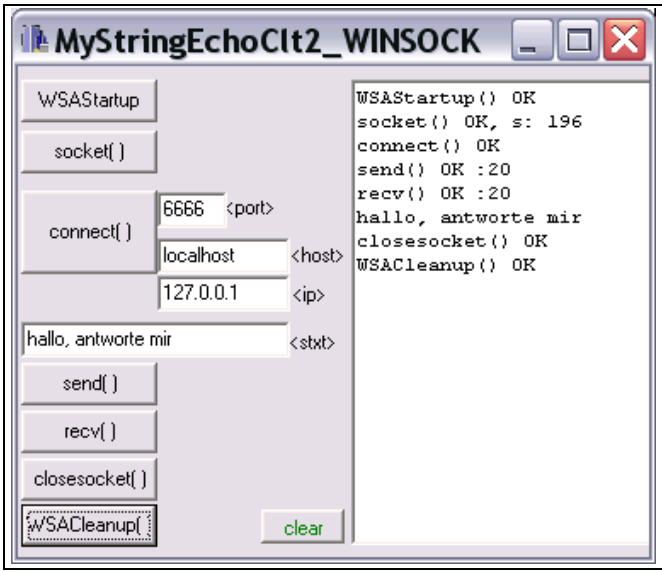
1.1 Ziel des Projektes

- Es soll ein String-Echo-Clt auf WINSOCK aufgesetzt werden.
- Der Client sendet an Server einen Text, der in Großbuchstaben gewandelt zurück gesendet wird.
- Der Server soll auf einem beliebigen Host und Port laufen, der Sendetext soll variierbar sein.

1.2 Voraussetzungen

Ein StringEchoServer läuft auf dem Teacher-Host "tc00-060401.it.htwm.de" oder lokal auf ihrem Host!

1.3 Realisierung des Projektes MyStringEchoClt2_WINSOCK



- Erzeugen Sie ein neues Projekt “MyStringEchoClt2_WINSOCK“ im gleichnamigen Order.
- Editieren Sie die Oberfläche entsprechend der nebenstehenden Abbildung.
- Erzeugen Sie Schritt für Schritt den Programmcode, entsprechend dem Beispiel-Code.

```
#include <vcl.h>
#include <winsock.h> //erforderlich
#include <stdio.h>   //erforderlich z.B. für putchar()
#include <string>
#pragma hdrstop
#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int s;          //socket descriptor
int len;        //für send() und recv()
char buf[4095]; //für send() und recv()
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::WSAStartup_Click(TObject *Sender)
{//===(1)Am API anmelden
 WSADATA wsaD;
  if  (WSAStartup (MAKEWORD(2,0),&wsaD) ==0) //MAKEWORD(maj,min)
      RichEdit1->Lines->Add("WSAStartup() OK");
  else
      RichEdit1->Lines->Add("WSAStartup() ERR");
}
void __fastcall TForm1::socket_Click(TObject *Sender)
```

```
{//===(2)Socket errichten
  s = socket(AF_INET,SOCK_STREAM,0);
  if (s!=0) RichEdit1->Lines->Add("socket() OK, s: "+IntToStr(s));
  else      RichEdit1->Lines->Add("socket() ERR");
}
//---------------------------------------------------------------------------
void __fastcall TForm1::connect_Click(TObject *Sender)
{//===(3a)sAddr mit Adressenfamilie und Port laden
   struct sockaddr_in  sAddr;//serveradresse
   sAddr.sin_family=AF_INET;
   sAddr.sin_port=htons(port->Text.ToInt());
 //===(3a)Name auf IP-Adresse auflösen und in sAddr schreiben
   if (hostname->Text!="")
       {
        struct hostent* hAddr = gethostbyname(hostname->Text.c_str());
        sAddr.sin_addr=(*(struct in_addr*) hAddr->h_addr);
        ip->Text=inet_ntoa(*(struct in_addr*) hAddr->h_addr);
       }
       else RichEdit1->Lines->Add("ERR bitte  <host> angeben");
 //===(4)Verbindung herstellen
      if (connect (s,(sockaddr*)&sAddr,sizeof(sockaddr))==0)
       RichEdit1->Lines->Add("connect() OK");
      else
       RichEdit1->Lines->Add("connect() ERR: "+IntToStr(WSAGetLastError()));
}
//---------------------------------------------------------------------------
void __fastcall TForm1::recv_Click(TObject *Sender)
{//===(5)Daten aus dem Socket lesen
   len = recv(s, buf, sizeof(buf),0);
   if ((len!=0)&& (len!=SOCKET_ERROR))
   {   RichEdit1->Lines->Add("recv() OK :"+IntToStr(len));
       buf[len]=0;
       RichEdit1->Lines->Add(buf);
   }
}
//---------------------------------------------------------------------------
void __fastcall TForm1::send_Click(TObject *Sender)
{//===(5)Daten in den Socket schreiben
      char *buf= stxt->Text.c_str();
      len =send (s,buf,strlen(buf),0);
      if (len!=SOCKET_ERROR)
        RichEdit1->Lines->Add("send() OK :"+IntToStr(strlen(buf)));
      else
        RichEdit1->Lines->Add("connect() ERR: "+IntToStr(WSAGetLastError()));
}
//---------------------------------------------------------------------------
void __fastcall TForm1::closesocket_Click(TObject *Sender)
{//===(6)Verbindung abbauen
   if (closesocket(s)!=SOCKET_ERROR) RichEdit1->Lines->Add("closesocket() OK");
   else
       RichEdit1->Lines->Add("closesocket() ERR: "+IntToStr(WSAGetLastError()));
}
//---------------------------------------------------------------------------
void __fastcall TForm1::WSACleanup_Click(TObject *Sender)
{//===(7)Am API abmelden
     WSACleanup();
     RichEdit1->Lines->Add("WSACleanup() OK");
}
//---------------------------------------------------------------------------
void __fastcall TForm1::clear_Click(TObject *Sender)
{
  RichEdit1->Clear();
}
```