

Jacek Matulewski, Sławek Orłowski

<http://www.phys.uni.torun.pl/~jacek/>

Delphi 7 + Indy 10

Przykłady prostych aplikacji sieciowych

Wersja α

Ćwiczenia

Toruń, 20 grudnia 2005

Najnowsza wersja tego dokumentu znajduje się pod adresem
http://www.phys.uni.torun.pl/~jacek/dydaktyka/rad/rad4_indy_delphi.pdf

Źródła opisanych w tym dokumencie programów znajdują się pod adresem
http://www.phys.uni.torun.pl/~jacek/dydaktyka/rad/rad4_indy.zip

Spis treści

Spis treści	2
Porty TCP. Skanowanie portów	3
Ćwiczenie 1 TCP - WWW	3
Ćwiczenie 2 Obsługa wyjątków	3
Ćwiczenie 3 Skanowanie portów	3
Klient FTP	5
Ćwiczenie 4 Klient FTP – przyczołek	5
Ćwiczenie 5 Dodać obsługę błędów do metody Button1Click	6
Ćwiczenie 6 (samodzielne) Klient FTP – sprawdzanie przy zamknięciu aplikacji, czy połączenie aktywne (pytanie o rozłączenie)	6
Ćwiczenie 7 Pobieranie listy plików – polecenie LIST – metoda TIdFTP.List	6
Ćwiczenie 8 Parserowanie listy plików	7
Ćwiczenie 9 „Eksploracja” serwera – zmiana bieżącego katalogu	8
Ćwiczenie 10 Przejście do katalogu nadrzędnego	9
Ćwiczenie 11 Pobieranie zaznaczonego pliku	9
Ćwiczenie 12 Ściąganie po dwukrotnym kliknięciu myszą listy	10
Ćwiczenie 13 Odświeżanie formy podczas pracy	11
Ćwiczenie 14 Wysyłanie pliku	11
Ćwiczenie 15 Kasowanie pliku z serwera	12
Ćwiczenie 16 Tworzenie katalogów na serwerze	12
Ćwiczenie 17 Kasowanie katalogów na serwerze	13
Inne	13
Ćwiczenie 18 Pobieranie plików przez HTTP	13
Ćwiczenie 19 PING	14

Porty TCP. Skanowanie portów

Ćwiczenie 1 TCP - WWW

1. Zakładka *IndyClients*
2. Na formę: *IdTCPClient*
 - a) *Host* = www.helion.pl
 - b) *Port* = 80
3. Na formę: *TButton*
4. Domyślna metoda zdarzeniowa:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    IdTCPClient1.Connect;
    ShowMessage('Połączony');
    IdTCPClient1.Disconnect;
    ShowMessage('Rozłączony');
end;
```

W przypadku przechwycenia wyjątku pojawi się okno z prośbą o wskazanie modułu klasy wyjątku (np. *IdException.pas*), który znajduje się w katalogu instalacyjnym Indy 10 for Delphi (np. *C:\Program Files\Indy 10 for Delphi\Source\System\IdException.pas*). Warto zaznaczyć opcję *Add directory to Debug Source Path*.

Ćwiczenie 2 Obsługa wyjątków

1. Dodajemy do uses: *IdException*.
2. Zmieniamy metodę zdarzeniową:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    try
        IdTCPClient1.Connect;
        ShowMessage('Połączony');
        IdTCPClient1.Disconnect;
        ShowMessage('Rozłączony');
    except
        on e :EIdException do ShowMessage('Błąd Indy: '+e.Message);
    end;
end;
```

Ćwiczenie 3 Skanowanie portów

1. Nowy projekt

2. Do formy dodajemy `IdTCPClient`
 - a) `ConnectTimeout = 3000` (3 sekundy)
3. Do formy dodajemy `TEdit`
 - a) `Text = www.helion.pl`
4. Dodajemy `TListBox`
5. Dodajemy przycisk `TButton`
 - a) `Caption = Połącz`
6. Domyślna metoda zdarzeniowa

```

procedure TForm1.Button1Click(Sender: TObject);
const
    porty :array[0..20] of Integer =
(7,20,21,22,23,25,53,79,80,110,117,137,143,161,220,443,531,1550,6000,8080,8888);
    protokoly :array[0..20] of String = ('echo','ftp-
data','ftp','ssh','telnet','smtp','dns','finger','http','pop3','uucp','netbios','imap','
snmp','imap3','ssl','irc','gg','x11','http-proxy','napster');
var
    i :Integer;
begin
    Edit1.Enabled:=False;
    Button1.Enabled:=False;

    ListBox1.Items.Clear();
    ListBox1.Items.Add('Sprawdzam porty komputera '+Edit1.Text);
    ListBox1.Items.Add('Liczba portów: '+IntToStr(Length(porty)));
    ListBox1.Items.Add('Skanowanie może potrwać dłuższą chwilę!');

    for i:=0 to Length(porty)-1 do
        begin
            Self.Refresh;
            try
                IdTCPClient1.Host:=Edit1.Text;
                IdTCPClient1.Port:=porty[i];
                IdTCPClient1.Connect();
                IdTCPClient1.Disconnect();
                ListBox1.Items.Add('Port '+IntToStr(porty[i])+' ('+protokoly[i]+' ) jest
otwarty');
            except
                on ex :EIdException do
                    ListBox1.Items.Add('Port '+IntToStr(porty[i])+' ('+protokoly[i]+' )
jest zamknięty ('+ex.Message+' )');
            end;
            ListBox1.ItemIndex:=ListBox1.Items.Count-1;
        end;

    ListBox1.Items.Add('Skanowanie zakończone. ');
    ListBox1.ItemIndex:=ListBox1.Items.Count-1;

```

```
Edit1.Enabled:=True;  
Button1.Enabled:=True;  
end;
```

Program należy uruchomić poza środowiskiem Delphi (albo bez debugowania), aby zapobiec przechwytywaniu wyjątków przez wbudowany debugger.

Klient FTP

Ćwiczenie 4 Klient FTP – przyczőłek

Ideologia FTP (2 x TCP). Tryb aktywny i pasywny

1. Nowy projekt
2. Do formy dodajemy `IdFTP` z zakładki *Indy Clients*.
 - a) `Host` = sunsite.icm.edu.pl
 - b) `TransferType` = `ttBinary`
 - c) `Username` = anonymous

3. Do formy dodajemy przycisk `TButton`

- a) `Caption` = Połącz

4. Domyślna metoda zdarzeniowa

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  IdFTP1.Connect;  
  ShowMessage('Połączony z '+IdFTP1.Username+'@'+IdFTP1.Host+'!'+#13+  
    'Katalog: '+IdFTP1.RetrieveCurrentDir);  
  Caption:=IdFTP1.Username+'@'+IdFTP1.Host;  
end;
```

5. Zmiana `IdFTP1.Passive` = True !!! (w zależności od ustawień lokalnego firewalla bez Passive może nie udać się nawiązać połączenia)

6. Przycisk `Button2`

- a) `Caption` = Rozłącz

7. Domyślna metoda zdarzeniowa

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
  IdFTP1.Disconnect;  
  ShowMessage('Rozłączony!');  
  Caption:='Rozłączony';  
end;
```

Do tej pory połączenie jedynie na porcie 21 (żadne dane nie zostały przesłane).

Ćwiczenie 5

Dodać obsługę błędów do metody Button1Click

Modyfikujemy metodę Button1Click:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
try
    IdFTP1.Connect;

    ShowMessage('Połączony z '+IdFTP1.Username+'@'+IdFTP1.Host+'!'+#13+'Katalog: '+IdFTP1.RetrieveCurrentDir);

    Caption:=IdFTP1.Username+'@'+IdFTP1.Host;
except
    on E: Exception do MessageBox(Handle, 'Błąd przy próbie połączenia', 'Błąd', MB_OK+MB_ICONERROR);
end;
end;
```

Ćwiczenie 6 (samodzielne)

Klient FTP – sprawdzanie przy zamknięciu aplikacji, czy połączenie aktywne (pytanie o rozłączenie)

Tworzymy metodę zdarzeniową do zdarzenia `CloseQuery` formy:

```
procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
if IdFTP1.Connected then
begin
    if MessageBox(Handle, 'Połączenie z serwerem jest aktywne.'+#13+'Kliknięcie OK spowoduje jego zamknięcie i zakończenie działania aplikacji.', 'Uwaga!', MB_OKCANCEL+MB_ICONWARNING)=ID_OK then
        Button2Click(Sender)
    else
        CanClose:=False;
end;
end;
```

Ćwiczenie 7

Pobieranie listy plików – polecenie LIST – metoda TIdFTP.List

1. Do formy dodajemy listę `TListBox`
2. Do formy dodajemy przycisk `TButton`
 - a) `Caption` = Zawartość katalogu
3. Domyślna metoda zdarzeniowa:

```
procedure TForm1.Button3Click(Sender: TObject);
begin
if not IdFTP1.Connected then
begin
    MessageBox(Handle, 'Brak połączenia', 'Błąd', MB_OK+MB_ICONERROR);
    Exit;
end;
```

```

    IdFTP1.List(ListBox1.Items);
end;

```

4. Przeciążona wersje metody List: `IdFTP1.List(ListBox1.Items, '', False);`.

Ćwiczenie 8 Parserowanie listy plików

- Dodajemy nowy przycisk
 - `Caption` = Zawartość katalogu (parser)
- Do `uses` dodajemy `IdFTPList` (są tam: klasa `TIdDirItemType` i stałe `ditFile`, `ditDirectory` i `ditSymbolicLink`, `ditSymbolicLinkDir`);
- Do sekcji `type` modułu dodajemy definicje typu wyliczeniowego oraz struktury mogącej przechowywać informacje o pozycjach w katalogu:

```

type
  TTypElementu = (tePlik, teKatalog, teLink);
  TElementKatalogu = record
    Nazwa :String;
    Typ :TTypElementu;
  end;

```

4. W deklaracji klasy umieszczamy pole – dynamiczną tablicę dla zdefiniowanego powyżej typu:

```

private
  ZawartoscKatalogu :array of TElementKatalogu;

```

5. Domyślna metoda zdarzeniowa:

```

procedure TForm1.Button4Click(Sender: TObject);
const pozycja=47;
var
  i,l :Integer;
  s :String;
  katalog,link :Boolean;
  element :TElementKatalogu;
begin
  if not IdFTP1.Connected then
    begin
      MessageBox(Handle, 'Brak połączenia', 'Błąd', MB_OK+MB_ICONERROR);
      Exit;
    end;

  ListBox1.Clear;
  SetLength(ZawartoscKatalogu,0);
  IdFTP1.List('.',True); //bieżący katalog

  //Wlasny parser do sunsite
  for i:=1 to IdFTP1.ListResult.Count-1 do
    begin
      s:=IdFTP1.ListResult[i];
      katalog:=(s[1]='d');

```

```

link:=(s[1]='l');
s:=Copy(s,pozycja+1,Length(s)-pozycja); //usuwanie informacji
s:=Trim(s); //na wszelki wypadek
if Pos('->',s)>0 then s:=Copy(s,1,Pos('->',s)-2);
l:=Length(ZawartoscKatalogu);
SetLength(ZawartoscKatalogu,l+1);
element.Nazwa:=s;
element.Typ:=tePlik;
if katalog then
begin
s:='['+s+']';
element.Typ:=teKatalog;
end;
if link then
begin
s:=s+'->';
element.Typ:=teLink;
end;
ZawartoscKatalogu[l]:=element;
ListBox1.Items.Add(s);
end;

Caption:=IdFTP1.Username+'@'+IdFTP1.Host+' Katalog: '+IdFTP1.RetrieveCurrentDir;
end;

```

Parser wbudowany (w Indy10 dla Win32 nie działa!)

```

ShowMessage('Ilość elementów w katalogu: '+IntToStr(IdFTP1.DirectoryListing.Count));
for i:=0 to IdFTP1.DirectoryListing.Count-1 do
begin
s:=IdFTP1.DirectoryListing.Items[i].FileName;
case IdFTP1.DirectoryListing.Items[i].ItemType of
ditFile: s:=s+' '+IntToStr(IdFTP1.DirectoryListing.Items[i].Size);
ditDirectory: s:='['+s+']';
ditSymbolicLink,ditSymbolicLinkDir: s:=s+'->';
end;
ListBox1.Items.Add(s);
end;

```

Ćwiczenie 9 „Eksploracja” serwera – zmiana bieżącego katalogu

Zareagujemy na kliknięcie listy. Tworzymy metodę do `OnDb1Click`:

```

procedure TForm1.ListBox1Db1Click(Sender: TObject);
var pozycja :Integer;
begin

```



```

pozycja:=ListBox1.ItemIndex;
//ShowMessage(IntToStr(pozycja)+' - '+ZawartoscKatalogu[pozycja].Nazwa);
case ZawartoscKatalogu[pozycja].Typ of
  teKatalog,teLink:
    begin
      try
        IdFTP1.ChangeDir(ZawartoscKatalogu[pozycja].Nazwa);
        Button4Click(Sender);
      except
        on E :Exception do MessageBox(Handle,PChar('Nie udało się zmienić katalogu
        ('+E.Message+'),''),'Błąd',MB_OK+MB_ICONERROR);
      end;
    end;
  end;
end;
end;

```

Ćwiczenie 10

Przejdźcie do katalogu nadrzędnego

1. Do formy dodajemy przycisk.
2. Zmieniamy jego etykietę na *Katalog nadrzędny*.
3. Tworzymy domyślny metodę nadrzędną:

```

procedure TForm1.Button5Click(Sender: TObject);
begin
  if not IdFTP1.Connected then
    begin
      MessageBox(Handle,'Brak połączenia','Błąd',MB_OK+MB_ICONERROR);
      Exit;
    end;

IdFTP1.ChangeDirUp;
  Button4Click(Sender);
end;

```

Ćwiczenie 11

Pobieranie zaznaczonego pliku

1. Na formie umieszczamy komponent **TSaveDialog**.
2. Ustalamy jego własności
 - a) **Title** := Wybierz położenie ściągane pliku
 - b) **Options.ofOverwritePrompt** := True
3. Dodajemy przycisk.
4. Zmieniamy jego etykietę na *Pobierz zaznaczony plik*
5. Tworzymy metodę zdarzeniową:

```

procedure TForm1.Button6Click(Sender: TObject);
var pozycja :Integer;

```

```

begin
if not IdFTP1.Connected then
begin
MessageBox(Handle, 'Brak połączenia', 'Błąd', MB_OK+MB_ICONERROR);
Exit;
end;

pozycja:=ListBox1.ItemIndex;
if (pozycja<0) and not (ZawartoscKatalogu[pozycja].Typ=tePlik) then Exit;
SaveDialog1.FileName:=ZawartoscKatalogu[pozycja].Nazwa;
if SaveDialog1.Execute then
begin
try
IdFTP1.Get(ZawartoscKatalogu[pozycja].Nazwa, SaveDialog1.FileName, True, False);
ShowMessage('Pobieranie pliku zakończone');
except
on E :Exception do MessageBox(Handle, PChar('Nie udało się pobrać pliku
('+E.Message+') '), 'Błąd', MB_OK+MB_ICONERROR);
end;
end;
end;
end;

```

Ćwiczenie 12

Ściąganie po dwukrotnym kliknięciu myszą listy

Uzupełniamy metodę `ListBox1Db1Click`:

```

procedure TForm1.ListBox1Db1Click(Sender: TObject);
var pozycja :Integer;
begin
pozycja:=ListBox1.ItemIndex;
//ShowMessage(IntToStr(pozycja)+' - '+ZawartoscKatalogu[pozycja].Nazwa);
case ZawartoscKatalogu[pozycja].Typ of
teKatalog, teLink:
begin
try
IdFTP1.ChangeDir(ZawartoscKatalogu[pozycja].Nazwa);
Button4Click(Sender);
except
on E :Exception do MessageBox(Handle, PChar('Nie udało się zmienić katalogu
('+E.Message+') '), 'Błąd', MB_OK+MB_ICONERROR);
end;
end;
tePlik:
if MessageBox(Handle, PChar('Czy pobrać plik
'+ZawartoscKatalogu[pozycja].Nazwa+'?'), 'Pobieranie', MB_YESNO+MB_ICONQUESTION)=ID_YES
then
Button6Click(Sender);
end;
end;

```

```
end;
```

Ćwiczenie 13 Odświeżanie formy podczas pracy

Tworzymy metodę zdarzeniową do `IdFTP1.OnWork` i umieszczamy w niej `Refresh`.

```
procedure TForm1.IdFTP1Work(ASender: TObject; AWorkMode: TWorkMode;  
    AWorkCount: Integer);  
begin  
    Refresh;  
    //Application.ProcessMessages;  
end;
```

Możemy też zająć się kolejką komunikatów (`Application.ProcessMessages`), ale tego bym odradzał, bo może to pozwolić użytkownikowi aplikacji nieźle namieszać.

Ćwiczenie 14 Wysyłanie pliku

1. Dodajemy komponent `TOpenDialog`.
 - a) `Title := Wybierz plik do wysłania`,
 - b) `Options.ofFileMustExist :=True`
2. Dodajemy przycisk z etykietą *Wyślij plik*

```
procedure TForm1.Button7Click(Sender: TObject);  
begin  
    if not IdFTP1.Connected then  
        begin  
            MessageBox(Handle, 'Brak połączenia', 'Błąd', MB_OK+MB_ICONERROR);  
            Exit;  
        end;  
  
    if OpenDialog1.Execute then  
        begin  
            try  
                IdFTP1.Put(OpenDialog1.FileName, SaveDialog1.FileName, False);  
                ShowMessage('Wysyłanie pliku zakończone');  
            except  
                on E :Exception do MessageBox(Handle, PChar('Nie udało się wysłać pliku ('+E.Message+') '), 'Błąd', MB_OK+MB_ICONERROR);  
            end;  
        end;  
  
    Button4Click(Sender);  
end;
```

Ćwiczenie 15

Kasowanie pliku z serwera

Dodajemy przycisk z etykietą *Usuń zaznaczony plik* i tworzymy domyślną metodę zdarzeniową:

```
procedure TForm1.Button8Click(Sender: TObject);
var pozycja :Integer;
begin
if not IdFTP1.Connected then
begin
MessageBox(Handle,'Brak połączenia','Błąd',MB_OK+MB_ICONERROR);
Exit;
end;

pozycja:=ListBox1.ItemIndex;
if (pozycja<0) and not (ZawartoscKatalogu[pozycja].Typ=tePlik) then Exit;
try
    IdFTP1.Delete(ZawartoscKatalogu[pozycja].Nazwa);
except
on E :Exception do MessageBox(Handle,PChar('Nie udało się usunąć pliku z serwera
('+E.Message+')'), 'Błąd',MB_OK+MB_ICONERROR);
end;

Button4Click(Sender);
end;
```

Ćwiczenie 16

Tworzenie katalogów na serwerze

Nowy przycisk z etykietą *Twórz katalog*:

```
procedure TForm1.Button9Click(Sender: TObject);
begin
if not IdFTP1.Connected then
begin
MessageBox(Handle,'Brak połączenia','Błąd',MB_OK+MB_ICONERROR);
Exit;
end;

if Edit1.Text='' then Exit;

try
    IdFTP1.MakeDir(Edit1.Text);
except
on E :Exception do MessageBox(Handle,PChar('Nie udało się stworzyć katalogu
('+E.Message+')'), 'Błąd',MB_OK+MB_ICONERROR);
end;

Button4Click(Sender);
```

```
end;
```

Ćwiczenie 17 Kasowanie katalogów na serwerze

Nowy przycisk z etykietą *Usuń zaznaczony katalog*:

```
procedure TForm1.Button10Click(Sender: TObject);
var pozycja :Integer;
begin
if not IdFTP1.Connected then
begin
MessageBox(Handle,'Brak połączenia','Błąd',MB_OK+MB_ICONERROR);
Exit;
end;

pozycja:=ListBox1.ItemIndex;
if (pozycja<0) and not (ZawartoscKatalogu[pozycja].Typ=teKatalog) then Exit;
try
IdFTP1.RemoveDir(ZawartoscKatalogu[pozycja].Nazwa);
except
on E :Exception do MessageBox(Handle,PChar('Nie udało się usunąć katalogu
('+E.Message+')'),'Błąd',MB_OK+MB_ICONERROR);
end;

Button4Click(Sender);
end;
```

Inne

Ćwiczenie 18 Pobieranie plików przez HTTP

1. Na formie umieszczamy komponenty **TEdit**, **TButton** (oba z palety *Standard*), **TSaveDialog** (paleta *Dialogs*) i **TIdHTTP** (paleta *Indy Clients*).
2. Tworzymy domyślną metodę zdarzeniową przycisku:

```
procedure TForm1.Button1Click(Sender: TObject);
var
strumien :TMemoryStream;
pozycja :Integer;
begin
pozycja:=LastDelimiter('/',Edit1.Text);
SaveDialog1.FileName:=Copy(Edit1.Text,pozycja+1,Length(Edit1.Text)-pozycja);
if SaveDialog1.Execute then
begin
try
```

```

try
    strumien:=TMemoryStream.Create;
    IdHTTP1.Get(Edit1.Text,strumien);
    strumien.SaveToFile(SaveDialog1.FileName);
    ShowMessage('Pobieranie zakoczono');
finally
    strumien.Free;
end;
except
    on E: Exception do ShowMessage('Błąd: '+E.Message);
end;
end;
end;

```

Ćwiczenie 19

PING

1. Na formie umieszczamy komponenty dwa komponenty `TEdit` oraz `TButton`, `TListBox` (*Standard*) i `TIdIcmpClient` (*Indy Clients*).
2. Domyślna metoda do `Button1` (bez obsługi błędów)

```

procedure TForm1.Button1Click(Sender: TObject);
var i: Integer;
begin
    IdIcmpClient1.Host:=Edit1.Text;
    ListBox1.Items.Add('Przesyłam 10 wiadomości o treści '+Edit2.Text);
    for i:=1 to 10 do IdIcmpClient1.Ping(Edit2.Text,i);
end;

```

3. Tworzymy metodę zdarzeniową do `IdIcmpClient1.OnReply`:

```

procedure TForm1.IdIcmpClient1Reply(ASender: TComponent;
    const AReplyStatus: TReplyStatus);
begin
    ListBox1.Items.Add(IntToStr(AReplyStatus.SequenceId)+' . Odpowiedź z
'+AReplyStatus.FromIpAddress+' odebrana w ciągu <='+IntToStr(AReplyStatus.TimeToLive)+' :
''+AReplyStatus.Msg+'", ilość bajtów='+IntToStr(AReplyStatus.BytesReceived));
end;

```