

ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ ПЕРЕДАЧІ ФАЙЛІВ

Сокети – це програмний інтерфейс, що забезпечує обмін інформацією між процесами.

Варто звернути увагу на те, що сокети – база для протоколів. Іншими словами, з їх допомогою можна самостійно створити (імітувати) POP, FTP, HTTP або будь-який інший протокол – навіть новий, свій власний.

Однією з основних переваг сокетного обміну інформацією в мережі можна назвати його гнучкість. Головний принцип роботи з сокетами полягає у відправці послідовності символів іншого комп'ютера, чи то повідомлення або цілий файл. Причому без “ручного” контролю правильності передачі (як під час роботи з СОМ-портами).

Для успішного з'єднання властивостями Port і Host компонента TClientSocket необхідно привласнити деякі значення.

У властивості Port потрібно вказати номер порту для підключення (1 – 65535, але краще брати з діапазону 1001 – 65535, тому що номери до 1000 можуть виявитися зайняті системними службами).

Host – хост-ім'я або IP-адреса комп'ютера, з яким потрібно з'єднатися. Наприклад, ua.delphi.com або 192.128.0.0.

Будемо розглядати сокет як чергу символів, що передаються з одного комп'ютера на інший. Відкрити сокет можна, викликавши метод Open (компонент TClientSocket) або привласнивши значення True властивості Active. Тут не зайвим буде поставити обробник виключення на випадок невдалого з'єднання.

Авторизація передбачає відправку серверу свого логіна і пароля, однак сама схема авторизації залежить від сервера (якщо він не вимагає введення цих даних, пункт можна пропустити).

Відправлення / прийом даних. Це якраз те, заради чого все затіяно. Від сервера залежить і те, через який протокол відбувається обмін даними.

Закриття сокета. По завершенню обміну даними потрібно закрити сокет, викликавши метод Close компонента TClientSocket або привласнивши значення False властивості Active.

Компонент TClientSocket має такі Властивості:

Active (тип: Boolean) – показник того, відкритий або закритий сокет. Відкрито – значення True, закритий – значення False. Доступно для запису.

Host (тип: string) – хост-ім'я комп'ютера, з яким потрібно з'єднатися.

Address (тип: string) – тут IP-адреса комп'ютера, з яким потрібно з'єднатися. На відміну від Host, тут може бути зазначений тільки IP. Різниця полягає в тому, що якщо в Host вказано буквене ім'я комп'ютера, то IP запитується у DNS.

Port (тип: Integer (Word)) – номер порта комп'ютера, з яким потрібно з'єднатися (1-65535).

Service (тип: string) – визначає службу, з портом якої відбудеться з'єднання (pop, ftp і т. д.). Можна сказати, що це якийсь довідник для зіставлення номера порту з будь-яким стандартним протоколом.

ClientType – містить тип передачі даних:
 ctBlocking – синхронна передача (OnRead і OnWrite не працюють). Синхронний тип підключення підходить для поточного обміну даними;
 ctNonBlocking – асинхронна передача (відправка / прийом даних може здійснюватися за допомогою подій OnRead і OnWrite).

Методи:

Open – відкриває сокет (привласнення властивості Active значення True).

Close – закриває сокет (привласнення властивості Active значення False).

Події:

OnConnect – виникає при установці підключення. У обробнику вже можна приступати до авторизації або відправлення / прийому даних.

OnConnecting – також виникає при підключенні. Відрізняється від OnConnect тим, що підключення ще не встановлено. Найчастіше використовується, наприклад, щоб оновити статус.

OnDisconnect – подія виникає при закритті сокета вашою програмою, віддаленим комп'ютером або через збій. OnError – подія виникає при помилці. Під час відкриття сокета ця подія не допоможе виявити помилку. Щоб уникнути появи повідомлення від Windows про помилку, краще подбати про внутрішню обробку виключень шляхом приміщення операторів відкриття в блок "try..except".

OnLookup – подія виникає при спробі отримати IP-адресу від DNS.

OnRead – подія виникає при відправці вам будь-яких даних віддаленим комп'ютером. При виклику OnRead можлива обробка отриманих даних.

OnWrite – подія виникає, коли вашій програмі дозволено писати дані в сокет.

Приклад:

{На самому початку необхідно створити форму TForm1 і екземпляр Form1, створити на формі кнопку TButton, при натисканні на яку буде викликатися процедура Button1Click – обробник події OnClick, і два поля TEdit, в які попередньо потрібно буде ввести хост-ім'я в один і порт в інший.

Обов'язково потрібно помістити компонент TClientSocket на форму!}

```
procedure Button1Click (Sender: TObject);
```

```
begin
```

```
{У Host і Port пишемо введені в формі значення}
```

```
ClientSocket1.Host: = Edit1.Text;
```

```
ClientSocket1.Port: = StrToInt (Edit2.Text);
```

```
{Відкриваємо сокет і намагаємося підключитися}
```

```
ClientSocket1.Open;
```

```
end;
```

```
procedure ClientSocket1Connect (Sender: TObject; Socket: TCustomWinSocket);
```

```
begin
```

```
{Якщо підключення пройшло успішно, сокет закриваємо, підключення обриваємо}
```

```
ClientSocket1.Close;
```

```
end;
```

Наприклад, деякі методи TClientSocket.Socket (TCustomWinSocket, TClientWinSocket):

SendBuf (var Buf; Count: Integer) – відправка буфера через сокет. Це може бути будь-який тип: простий Integer, структура (record) тощо.

Вказати буфер потрібно параметром Buf. У другому параметрі повинен бути зазначений розмір (у байтах) переданих даних (Count);

SendText (const S: string) – відправка текстового рядка;

SendStream (AStream: TStream) – відправка вмісту потоку будь-якого типу: файлового, з ОЗУ і т. д. Цей потік повинен бути відкритий.

Отже, у даному дослідженні з'ясовано, що сокети – база для протоколів. Найпростіша програма реалізації обміну інформацією дає можливість протестувати сокети та перевірити їх на наявність помилок.

Список використаної літератури

1. Гук М. А. Аппаратные средства локальных сетей: энциклопедия / М. А. Гук. – СПб : Питер, 2010. – 576 с.
2. Новиков Ю. В. Локальные сети: архитектура, алгоритмы, проектирование / Ю. В. Новиков, С. В. Кондратенко. – М. : ЭКОМ, 2011 – 312 с.
3. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов / В. Г. Олифер, Н. А. Олифер. – СПб : Питер, 2012. – 864 с.
4. Соловьёва Л. Ф. Сетевые технологии: учебник-практикум / Л. Ф. Соловьёва. – СПб. : БХВ-Петербург, 2014. – 416 с.

Анастасія Білівненко

СИСТЕМА ПІДГОТОВКИ МАЙБУТНІХ ПЕДАГОГІВ ЗА КОРДОНОМ: ДОСВІД НІМЕЦЬКОГО УНІВЕРСИТЕТУ У М. БІЛЕФЕЛЬД

Педагогічна освіта у Німеччині має багату історичну основу. Першими спробами створення спеціальних закладів для підготовки вчителів були педагогічні семінарії, засновані Августом Германом Франке у 1730 році. Їхньою особливістю було те, що в основу покладалася наукова підготовка майбутніх учителів, тоді як практична підготовка до уваги майже не бралась [1]. Але згодом Філантропісти, представники найпрогресивнішої педагогічної течії Німеччини, зробили наступний крок у справі підготовки майбутніх педагогів. Вони висунули такі вимоги спеціальної підготовки майбутніх вчителів, а саме, оригінальні методи викладання, ґрунтовне знання мов, практична діяльність вчителя, повага до учня як до особистості із власним баченням. У такій специфіці викладання відобразились найкращі сторони німецької школи.

Кінець XVIII – початок XIX ст. приніс у систему педагогічної освіти Німеччини радикальні зміни, пов'язані з ім'ям Гумбольдта та сформульованій ним класичної ідеї університету. Згідно цій ідеї, наукові дослідження у вищій школі поєднувалися з навчанням. Держава забезпечувала свободу тим, хто навчається, і одночасно тим, хто навчає. На першому плані стояли завдання розвитку науки, використання нових форм та методів навчання [3].