



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

**МУНИЦИПАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ ДЕТЕЙ
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ ДЕТЕЙ**

Специальность Мастер по обработке цифровой информации

Отдел Информационные технологии

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Программа Р2Р Чат Pandora.

Автор

(Подпись)

(Дата)

(Фамилия, инициалы)

Руководитель

(Подпись)

(Дата)

(Фамилия, инициалы)

ВОРОНЕЖ – 2018 г.

Оглавление

Введение	3
Постановка цели и задач	5
Код программы.....	6
Блок-схема.....	9
Описание переменных и компонентов	10
Отчет о тестировании	11
Заключение.....	12
Список литературы	13

Введение

Свою курсовую работу я решил провести в среде Разработки Delphi 10.1 Berlin.

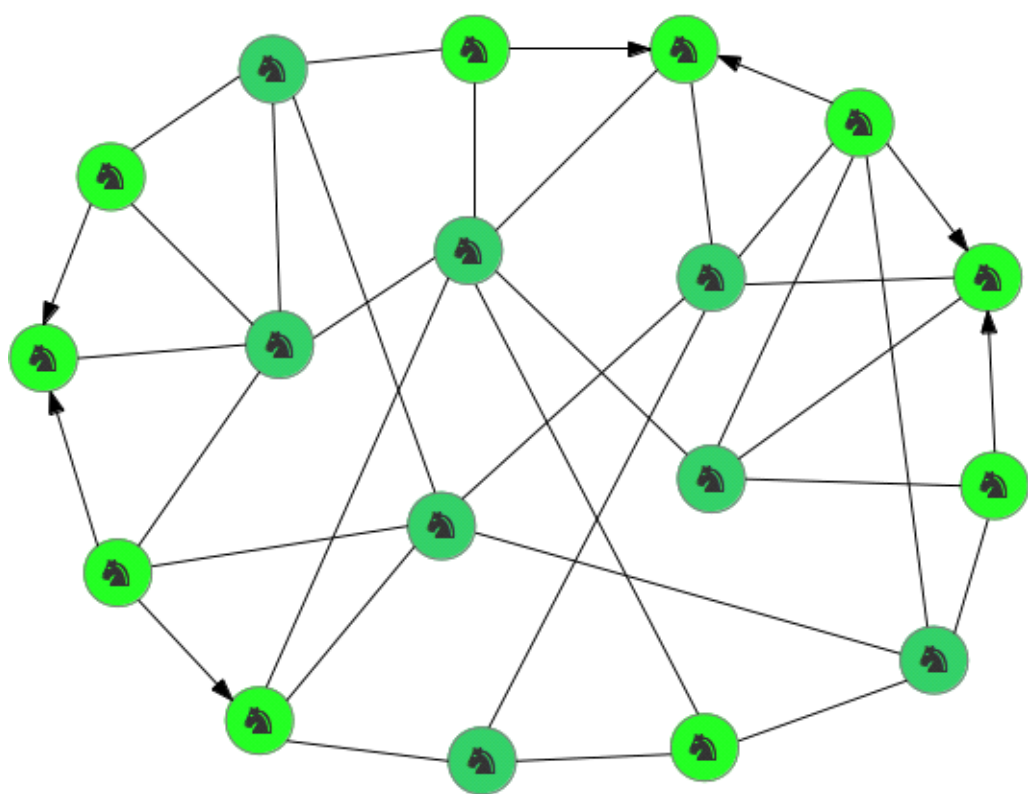
Зачем же нам чаты, мессенджеры, социальные сети? Чаты, мессенджеры, социальные сети изменили наше представление об общении в интернете и прочно вошли в нашу жизнь, и уходить не собираются. С помощью компьютерных сетей - мы узнаём новое, общаемся с близкими, участвуем в разработке разных проектов. Каждый человек имеет потребность общаться в современном мире - эту роль выполняют разные программы, но порой многие программы **централизованные и имеют ряд серьезных проблем. Централизованные сети подвержены утечкам данных, цензуре и выходу из строя. Поэтому** я решил создать свое ПО для общения в интернете построенное по технологии децентрализованной сети.

Структуры обычной сети:

Это традиционные социальные сети. Сервер принадлежит одному владельцу или группе людей. Пользователи полностью доверяют информацию владельцу и спецслужбам. Данные хранятся удаленно и централизованно на сервере. Такие сети подвержены отказу в обслуживании, взлому, массовому сливу данных.

Моя распределенная (p2p) сеть Pandora.

Сеть полностью децентрализованная. Невозможно вывести сеть из строя или взять её под контроль. Так как обмен данными происходит среди близких, друзей и вероятность утечек данных третьим лицам почти отсутствует. Пример общения через Pandora.



Постановка цели и задач

Цель: написать эффективный децентрализованный чат в среде программирования Delphi 10.1 Berlin.

Задачи:

- Разработать план действий по реализации децентрализованной сети для отправки сообщений.
- Реализовать простой интерфейс для удобного поднятия своего сервера и подключения к серверу и максимальной мощный функционал.
- Написать программный код для реализации обмена сообщениями.
- Протестировать программу на различных конфигурациях оборудования.
- Отладить программу, найти баги и исправить их.
- Представить готовую и стабильную реализацию пирингового чата.

Код программы

```
unit Unit1;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, System.Win.ScktComp, Vcl.StdCtrls,
  Vcl.Menus, Vcl.ComCtrls, Vcl.ExtCtrls, ShellApi;

type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    ClientSocket1: TClientSocket;
    MainMenu1: TMainMenu;
    Help1: TMenuItem;
    ServerSocket1: TServerSocket;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    ListBox1: TListBox;
    ListBox2: TListBox;
    Button6: TButton;
    Label1: TLabel;
    Label2: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure ClientSocket1Read(Sender: TObject; Socket: TCustomWinSocket);
    procedure Edit1KeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
    procedure Help1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure ServerSocket1ClientRead(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocket1ClientConnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocket1ClientDisconnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure Button4Click(Sender: TObject);
    procedure ServerSocket1ClientError(Sender: TObject;
      Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
      var ErrorCode: Integer);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure ServerSocket1Accept(Sender: TObject; Socket: TCustomWinSocket);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
```

```

Form1: TForm1;
Login:string;
Host:string;
Port:string;
ServerLocalAddress:string;
ServerLocalHost:string;
implementation
{$R *.dfm}
uses Unit2;
procedure TForm1.Button1Click(Sender: TObject);
begin
    ClientSocket1.Socket.SendText(Login+'.'+Edit1.Text);
    Edit1.Clear;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
    Host:=InputBox('Ip','Host:','127.0.0.1');
    Port:=InputBox('Port','Port:','5050');
    ClientSocket1.Port:=StrToInt(Port); //Port
    ClientSocket1.Host:=Host;
    ClientSocket1.Active:=True;
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
    ServerSocket1.Active:=False; // ServerSocket1.close;
    Port:=InputBox('Port','Port:','5050');
    ServerSocket1.Port:=StrToInt(Port);
    ServerSocket1.Active:=True; // ServerSocket1.close;
    if ServerSocket1.Active then
    begin
        ListBox1.Items.Add('Server Create!');
        ServerLocalAddress:=ServerSocket1.Socket.LocalAddress;
        ServerLocalHost:=ServerSocket1.Socket.LocalHost;
        ListBox1.Items.Add('Возвращает имя сервера в сети: '+ServerLocalHost);
        ListBox1.Items.Add('Возвращает IP сервера.'+ServerLocalAddress);
    end
    else ListBox1.Items.Add('Server Failed!');
end;
procedure TForm1.Button4Click(Sender: TObject);
begin
    ShellExecute(Application.Handle,PChar('open'),PChar('https://cloud.mail.ru/home/'),PChar(0),nil,SW_NORMAL);
end;
procedure TForm1.Button5Click(Sender: TObject);
begin
    ListBox1.Items.Clear;
end;
procedure TForm1.Button6Click(Sender: TObject);
begin
    ServerSocket1.Active:=True; // ServerSocket1.close;
    ListBox1.Items.Add('Server Open!');
end;
procedure TForm1.Button7Click(Sender: TObject);
begin
    Port:=InputBox('Port','Port:','5050');
    ServerSocket1.Port:=StrToInt(Port);
    ServerSocket1.Active:=True; // ServerSocket1.Open;
    if ServerSocket1.Active then
    begin
        ListBox1.Items.Add('Server Open!');
    end
    else ListBox1.Items.Add('Server Not Open!');
end;
end;

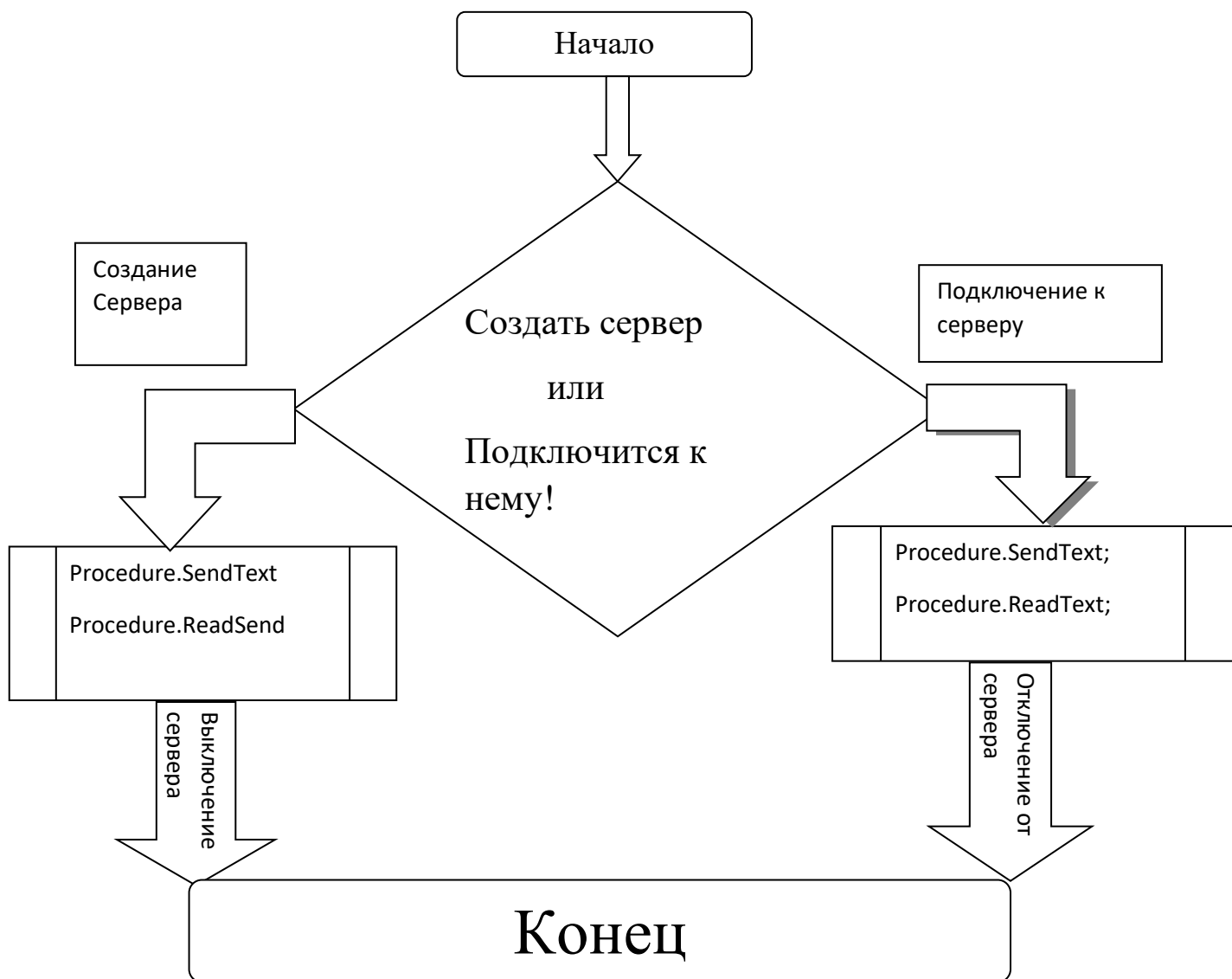
```

```

procedure TForm1.ClientSocket1Read(Sender: TObject; Socket: TCustomWinSocket);//Ответ от сервера
var Mes:String;
begin
  Mes:=Socket.ReceiveText;
  ListBox2.Items.Add(Mes); //
end;
procedure TForm1.Edit1KeyDown(Sender: TObject; var Key: Word;  //Entert=13 delphi
  Shift: TShiftState);
begin
  if Key=13 then
  begin
    Button1.Click;
  end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
  Login:=InputBox('Name','Login:','User1');
end;
procedure TForm1.FormDestroy(Sender: TObject);
begin
  ServerSocket1.Close; // CClose сокет
  ClientSocket1.Close; // Закрываем сокет
end;
procedure TForm1.Help1Click(Sender: TObject);
begin
  Form2.Show;
end;
procedure TForm1.ServerSocket1Accept(Sender: TObject; Socket: TCustomWinSocket);
begin //возникает, когда сервер принимает клиента или отказывает ему в соединении;
end;
procedure TForm1.ServerSocket1ClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  ListBox1.Items.Add('Connected:.'+Socket.RemoteAddress+');' );
end;
procedure TForm1.ServerSocket1ClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  ListBox1.Items.Add('Disconnect:.'+Socket.RemoteAddress+');' );
end;
procedure TForm1.ServerSocket1ClientError(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent; var ErrorCode: Integer);
begin
  {Произошла ошибка - выводим ее код}
  ListBox1.Items.Insert(0, 'Client error.Code = ' + IntToStr(ErrorCode));
end;
procedure TForm1.ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
var
  I:integer;
  Mes:String;
begin
  Mes:=Socket.ReceiveText;
  ListBox1.Items.Add('Talk:'+Mes+');');
  for I := 0 to ServerSocket1.Socket.ActiveConnections-1 do
  begin
    ServerSocket1.Socket.Connections[I].SendText(Mes);
  end;end;end.

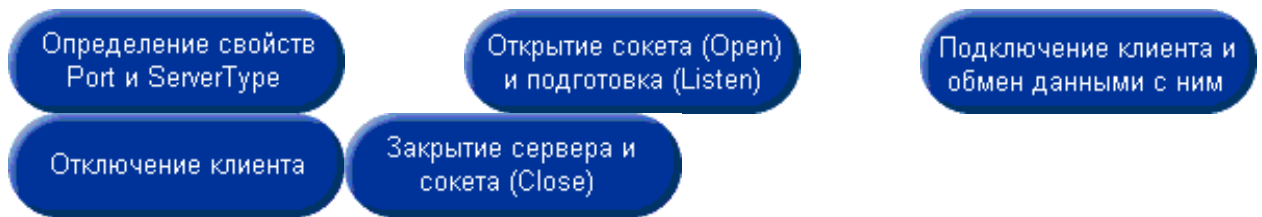
```


Блок-схема



Описание переменных и компонентов

TServerSocket - сетевой компонент. Он используется для создания сервера.



TServerSocket имеет свои методы, события, свойства.

Active - показатель того, активен в данный момент сервер, или нет. Т.е., фактически, значение True указывает на то, что сервер работает и готов к приему клиентов, а False - сервер выключен. Чтобы запустить сервер, нужно просто присвоить этому свойству значение True. Тип: Boolean;

Port - номер порта для установления соединений с клиентами. Порт у сервера и у клиентов должны быть одинаковыми. Рекомендуются значения от 1025 до 65535, т.к. от 1 до 1024 - могут быть заняты системой. Тип: Integer;

События:

OnClientConnect - возникает, когда клиент установил соединение.

OnClientDisconnect - возникает, когда клиент отсоединился.

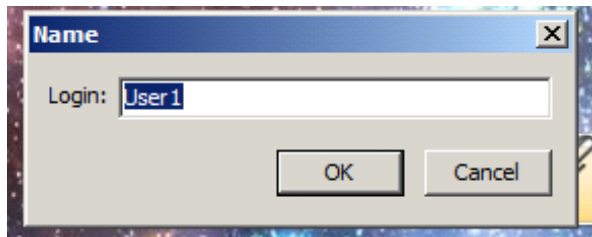
OnClientRead - возникает, когда клиент передал серверу данные.

OnClientWrite - возникает, когда сервер может отправлять данные клиенту по сокету.

TClientSocket - сетевой компонент, который подключается к созданному серверу и общается с другими клиентами. TClientSocket имеет такие же базовые команды, как и TServerSocket.

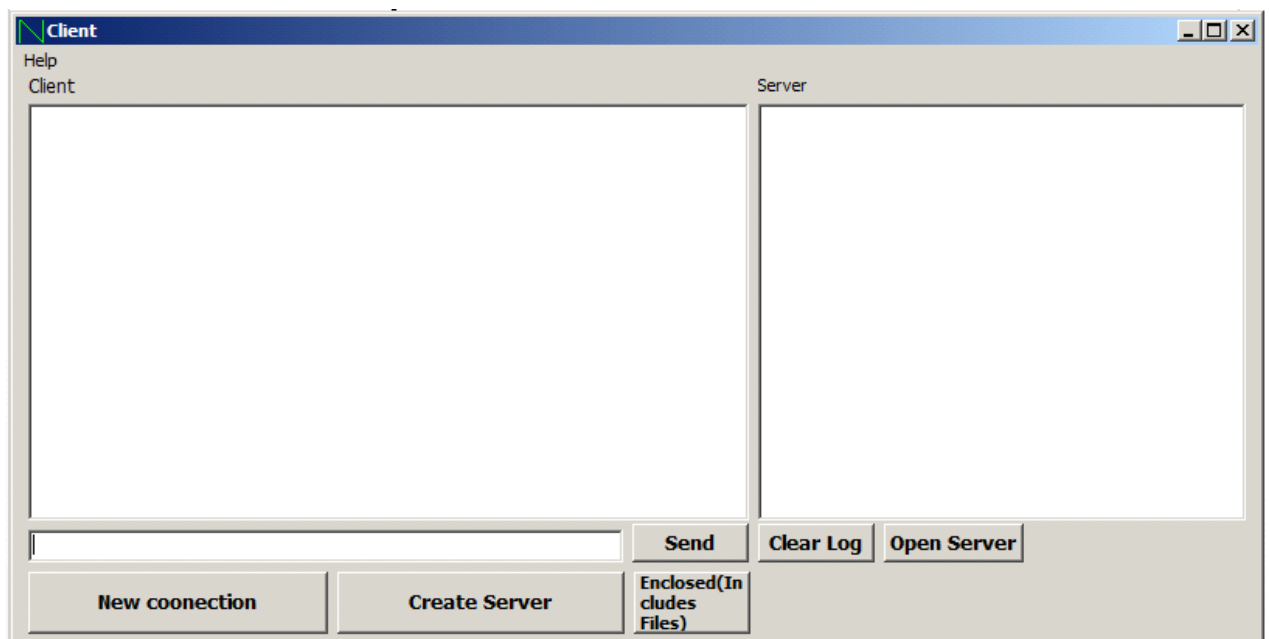
Отчет о тестировании

При запуске программы появляется следующее окно:



В указанную форму надо ввести свой никнейм, который будет отображаться другим пользователям.

Потом откроется уже пользовательское окно. В нем надо нажать на кнопку New Connection, чтобы подключиться к серверу или создать свой сервер для себя и своих друзей. Кнопка Clear Log очищает поле вывода сервера, а кнопка Open Server помогает заново открыть сервер если он не открылся по техническим причинам. Кнопка Enclose помогает отправить документ своему собеседнику.



Вкладка Help нужна, чтобы помочь пользователю в использование моего продукта.

Заключение

В ходе исследовательской работы я достиг своей цели и реализовал все поставленные задачи.

В процессе написания программы возникли проблемы с отправкой файлов. Это помогло решить облачный файлообменник.

В будущем планирую расширить и изменить функционал чата. Сделать шифрование сообщений. Добавить отправку смайликов, поддержку разных форматов файлов. А также дополнить вкладку Help, чтобы упростить и объяснить пользователю, как работает мой продукт.

Список литературы

1. Фундаментальные алгоритмы и структуры данных
2. Фленов М.Е. «Библия Delphi». 2015г. с.880 (4-е издание)
3. Осипов, Д. Delphi. Профессиональное программирование
4. Культин Н.Б. «Основы программирования в Embarcadero Delphi»: Интернет-издание, 2015.
5. Интернет ресурс: "<http://www.cyberforum.ru/delphi/>
6. <http://www.delphimaster.ru/articles/sockets/>
7. http://www.delphisources.ru/pages/faq/base/sockets_in_delphi.htm