

Затверджено

482.362. 6050102-02 12 09-1

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

Інститут фізико-технічних та комп'ютерних наук  
Кафедра комп'ютерних систем та мереж

**ВАРІАНТ 9**

**482.362. 6050102-02 12 09-1**

(Текст програми)

Сторінок 11

2015

**АНОТАЦІЯ**

В даному програмному документі представлено текст програми, розробленої в середовищі Embarcadero Delphi XE.

**ТЕКСТ ПРОГРАМИ**

```
unit MainUnit;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, ComCtrls,
    DBXJSON,
    ShellAPI;

type

    TMainForm = class(TForm)
        NewLecturerGroup: TGroupBox;
        NameEdit: TEdit;
        GradeEdit: TEdit;
        OfficeEdit: TEdit;
        AddNewLecturerBtn: TButton;
        NewCourseGroup: TGroupBox;
        AddLecturerIDEdit: TEdit;
        CourseTitleEdit: TEdit;
        AddNewCourseBtn: TButton;
        RemoveLecturerGroup: TGroupBox;
        RemoveLecturerIDEdit: TEdit;
        RemoveLecturerBtn: TButton;
        SaveJsonDialog: TSaveDialog;
        WorkWithJSONGroup: TGroupBox;
        ExportAsJSONBtn: TButton;
        ImportFromJSONBtn: TButton;
        OpenJsonDialog: TOpenDialog;
        LecturersListView: TListView;
        SalaryEdt: TEdit;
        PrintBtn: TButton;
        procedure AddNewLecturerBtnClick(Sender: TObject);
        procedure AddNewCourseBtnClick(Sender: TObject);
        procedure RemoveLecturerBtnClick(Sender: TObject);
        procedure ExportAsJSONBtnClick(Sender: TObject);
        procedure ImportFromJSONBtnClick(Sender: TObject);
        procedure PrintBtnClick(Sender: TObject);
    private
        { Private declarations }
    end;
```

482.362. 6050102-02 12 09-1

```

public
    { Public declarations }
end;

TLecturer = class(TObject)
public
    Name: string[40];
    Grade: string[40];
    Office: string[40];
    Courses: array of string;
    constructor Create(const Name, Grade, Office: string);
    constructor CreateFromJson(const JSONObject: TJSONObject); virtual;
    destructor Destroy;
    function ToJSON(): TJSONObject; virtual;
end;

TWage = class(TLecturer)
public
    Salary: Real;
    Sum: Real;
    constructor Create(const Name, Grade, Office: string; const Salary:
Real);
    constructor CreateFromJson(const JSONObject: TJSONObject); override;
    function ToJSON(): TJSONObject; override;
end;

var
    MainForm: TMainForm;
    Lecturers: array of TWage;

implementation

constructor TLecturer.Create(const Name, Grade, Office: string);
begin
    Self.Name := Name;
    Self.Grade := Grade;
    Self.Office := Office;
end;

constructor TLecturer.CreateFromJson(const JSONObject: TJSONObject);
var
    JSONCourses: TJSONArray;

```

```
Name, Grade, Office: string;
i: Integer;
begin
  try
    Name := JSONObject.Get('name').JsonValue.Value;
    Grade := JSONObject.Get('grade').JsonValue.Value;
    Office := JSONObject.Get('office').JsonValue.Value;
    Self.Create(Name, Grade, Office);

    JSONCourses := JSONObject.Get('courses').JsonValue as TJSONArray;

    SetLength(Self.Courses, JSONCourses.Size);
    for i := 0 to JSONCourses.Size - 1 do
      Self.Courses[i] := JSONCourses.Get(i).Value;
    except
      raise Exception.Create('Помилка парсингу JSON!');
    end;
  end;
end;

function TLecturer.ToJSON(): TJSONObject;
var
  JSONObject: TJSONObject;
  JSONArray: TJSONArray;
  i: Integer;
begin
  JSONObject := TJSONObject.Create;

  JSONObject.AddPair(TJSONPair.Create('name', Self.Name));
  JSONObject.AddPair(TJSONPair.Create('grade', Self.Grade));
  JSONObject.AddPair(TJSONPair.Create('office', Self.Office));

  JSONArray := TJSONArray.Create;
  for i := 0 to Length(Self.Courses) - 1 do
    begin
      JSONArray.Add(Self.Courses[i]);
    end;

    JSONObject.AddPair(TJSONPair.Create('courses', JSONArray));

  Result := JSONObject;
end;
```

```
destructor TLecturer.Destroy;
begin
    // Nothing to do
    inherited;
end;

constructor TWage.Create(const Name: string; const Grade: string;
    const Office: string; const Salary: Real);
begin
    inherited Create(Name, Grade, Office);
    Self.Salary := Salary;
    Self.Sum := 0;
end;

constructor TWage.CreateFromJson(const JSONObject: TJSONObject);
begin
    inherited CreateFromJson(JSONObject);

    try
        Self.Salary := StrToFloat(JSONObject.Get('salary').JsonValue.Value);
        Self.Sum := StrToFloat(JSONObject.Get('sum').JsonValue.Value);
    except
        raise Exception.Create('Помилка парсингу JSON.');
```

end;

```
end;

function TWage.ToJSON: TJSONObject;
var
    JSONObject: TJSONObject;
begin
    JSONObject := inherited ToJSON();

    JSONObject.AddPair(TJSONPair.Create('salary', FloatToStr(Self.Salary)));
    JSONObject.AddPair(TJSONPair.Create('sum', FloatToStr(Self.Sum)));

    Result := JSONObject;
end;

{$R *.dfm}
```

482.362. 6050102-02 12 09-1

```

procedure RefreshLecturersListBox;
var
  i, j: Integer;
  NewRow: TListItem;
  Courses: string;
begin
  MainForm.LecturersListView.Items.Clear;

  for i := 0 to Length(Lecturers) - 1 do
    begin
      Courses := '';
      for j := 0 to Length(Lecturers[i].Courses) - 2 do
        Courses := Courses + Lecturers[i].Courses[j] + ', ';
      if (Length(Lecturers[i].Courses) >= 1) then
        Courses := Courses + Lecturers[i].Courses
          [Length(Lecturers[i].Courses) - 1];

      NewRow := MainForm.LecturersListView.Items.Add;
      NewRow.Caption := Lecturers[i].Name;
      NewRow.SubItems.Add(Lecturers[i].Grade);
      NewRow.SubItems.Add(Lecturers[i].Office);
      NewRow.SubItems.Add(Courses);
      NewRow.SubItems.Add(FloatToStr(Lecturers[i].Salary));
      NewRow.SubItems.Add(FloatToStr(Lecturers[i].Sum));
    end;
  end;

procedure TMainForm.AddNewCourseBtnClick(Sender: TObject);
var
  LecturerID, Len: Integer;
begin
  LecturerID := StrToInt(AddLecturerIDEdit.Text);

  Len := Length(Lecturers[LecturerID].Courses);
  SetLength(Lecturers[LecturerID].Courses, Len + 1);

  Lecturers[LecturerID].Courses[Len] := CourseTitleEdit.Text;
  Lecturers[LecturerID].Sum := Length(Lecturers[LecturerID].Courses) *
    Lecturers[LecturerID].Salary;

  RefreshLecturersListBox;

```

```
end;
```

```
procedure TMainForm.AddNewLecturerBtnClick(Sender: TObject);
```

```
var
```

```
    NewLecturer: TWage;
```

```
    Name, Grade, Office, Salary: string;
```

```
    Len: Integer;
```

```
begin
```

```
    Name := NameEdit.Text;
```

```
    Grade := GradeEdit.Text;
```

```
    Office := OfficeEdit.Text;
```

```
    Salary := SalaryEdt.Text;
```

```
    Len := Length(Lecturers);
```

```
    SetLength(Lecturers, Len + 1);
```

```
    Lecturers[Len] := TWage.Create(Name, Grade, Office, StrToFloat(Salary));
```

```
    RefreshLecturersListBox;
```

```
end;
```

```
procedure TMainForm.ExportAsJSONBtnClick(Sender: TObject);
```

```
var
```

```
    JSONObject: TJSONObject;
```

```
    JSONArray: TJJSONArray;
```

```
    str: TStringList;
```

```
    i: Integer;
```

```
begin
```

```
    if SaveJsonDialog.Execute then
```

```
    begin
```

```
        try
```

```
            str := TStringList.Create;
```

```
            JSONObject := TJSONObject.Create;
```

```
            JSONArray := TJJSONArray.Create;
```

```
            for i := 0 to Length(Lecturers) - 1 do
```

```
            begin
```

```
                JSONArray.AddElement(Lecturers[i].ToJSON);
```

```
            end;
```

```
            JSONObject.AddPair(TJSONPair.Create('lecturers', JSONArray));
```



482.362. 6050102-02 12 09-1

```
        str.Add(JSONObject.ToString);
        str.SaveToFile(SaveJsonDialog.FileName);
    finally
        str.Free;
        JSONObject.Destroy;
    end;
end;
end;

procedure TMainForm.ImportFromJSONBtnClick(Sender: TObject);
var
    JSONObject: TJSONObject;
    JSONLecturers: TJSONArray;
    str: TStringList;
    i: Integer;
begin
    if OpenJsonDialog.Execute then
    begin
        try
            str := TStringList.Create;
            str.LoadFromFile(OpenJsonDialog.FileName);

            JSONObject := TJSONObject.ParseJSONValue(str.Text) as TJSONObject;

            JSONLecturers := JSONObject.Get('lecturers').JsonValue as TJSONArray;

            SetLength(Lecturers, JSONLecturers.Size);

            for i := 0 to JSONLecturers.Size - 1 do
                Lecturers[i] := TWage.CreateFromJson(JSONLecturers.Get(i)
                    as TJSONObject);

                RefreshLecturersListBox;
            finally
                str.Free;
            end;
        end;
    end;
end;

procedure TMainForm.PrintBtnClick(Sender: TObject);
var
```

482.362. 6050102-02 12 09-1

```

i, j: Integer;
Name, Grade, Office, Courses, Salary, Sum, str: String;
OutFile: TextFile;
begin
  try
    AssignFile(OutFile, 'print.txt');
    Rewrite(OutFile);

    for i := 0 to LecturersListView.Items.Count - 1 do
      begin
        Name := LecturersListView.Items[i].Caption;
        Grade := LecturersListView.Items[i].SubItems.Strings[0];
        Office := LecturersListView.Items[i].SubItems.Strings[1];
        Courses := LecturersListView.Items[i].SubItems.Strings[2];
        Salary := LecturersListView.Items[i].SubItems.Strings[3];
        Sum := LecturersListView.Items[i].SubItems.Strings[4];

        str := Format('%-30s', [Name]) + Format('%-10s', [Grade]) +
          Format('%-20s', [Office]) + Format('%-40s', [Courses]) +
          Format('%-10s', [Salary]) + Format('%10s', [Sum]);
        Writeln(OutFile, str);
      end;
    finally
      CloseFile(OutFile);
    end;

    ShellExecute(Handle, 'print', 'print.txt', nil, nil, SW_Hide);
  end;

  procedure TMainForm.RemoveLecturerBtnClick(Sender: TObject);
  var
    LecturerID, i: Integer;
  begin
    LecturerID := StrToInt(RemoveLecturerIDEdit.Text);

    Lecturers[LecturerID].Destroy;
    for i := LecturerID to Length(Lecturers) - 1 do
      begin
        Lecturers[i] := Lecturers[i + 1];
      end;
    end;
  end;

```

```
SetLength(Lecturers, Length(Lecturers) - 1);  
RefreshLecturersListBox;  
end;  
  
end.
```