**РУСЕНСКИ УНИВЕРСИТЕТ „АНГЕЛ КЪНЧЕВ“**

**КУРСОВА РАБОТА**

ПО

СОФТУЕРНО ИНЖЕНЕРСТВО

Студент:

Фак. Номер:

Група:

Специалност:

Дата: Проверил:

# Съдържание

[Съдържание 2](#_Toc372370749)

[1. Задание 3](#_Toc372370750)

[2. Диаграми 3](#_Toc372370751)

[2.1 Delphi диаграма 3](#_Toc372370754)

[2.2 Use Case 4](#_Toc372370755)

[2.3 Sequence 5](#_Toc372370756)

[2.4 Activity 6](#_Toc372370757)

[2.5 Class 7](#_Toc372370758)

[3. Работа с програмата 8](#_Toc372370759)

[4. Код на програмата 9](#_Toc372370760)

[4.1 KMain.pas 10](#_Toc372370761)

[4.2 About.pas 12](#_Toc372370762)

# Задание

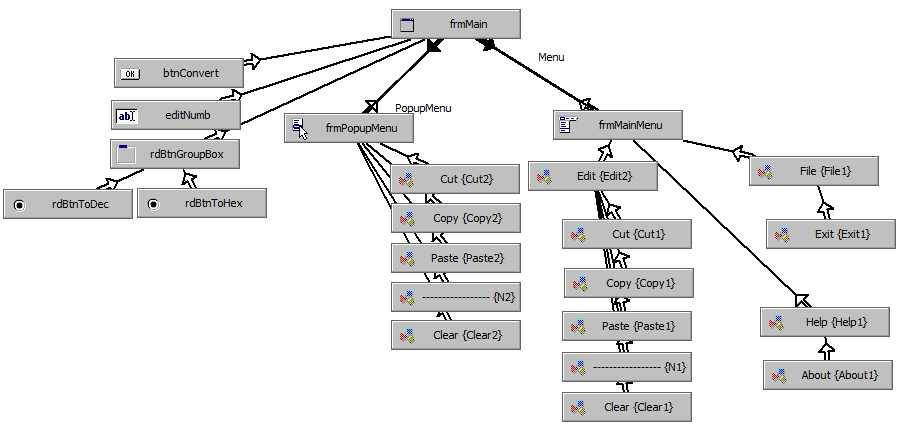
Да се проектира и реализира Windows приложение конвертор на десетични числа в шестнадесетична бройна система и обратно:

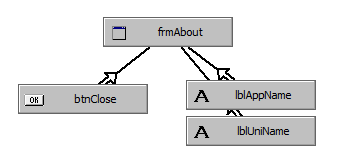
* Главно меню с елементи File и Edit;
* Контекстно меню с елементи Cut, Copy, Paste и Clear;
* About форма;
* Автоматично записване на извършените операции във файл.

# Диаграми

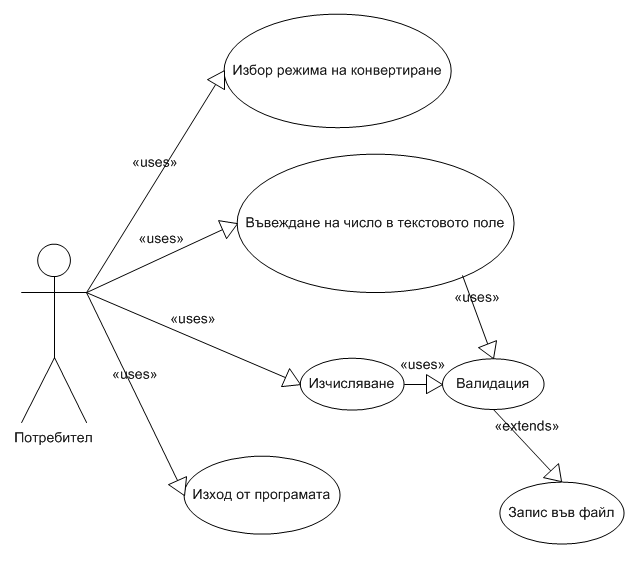


## Delphi диаграма

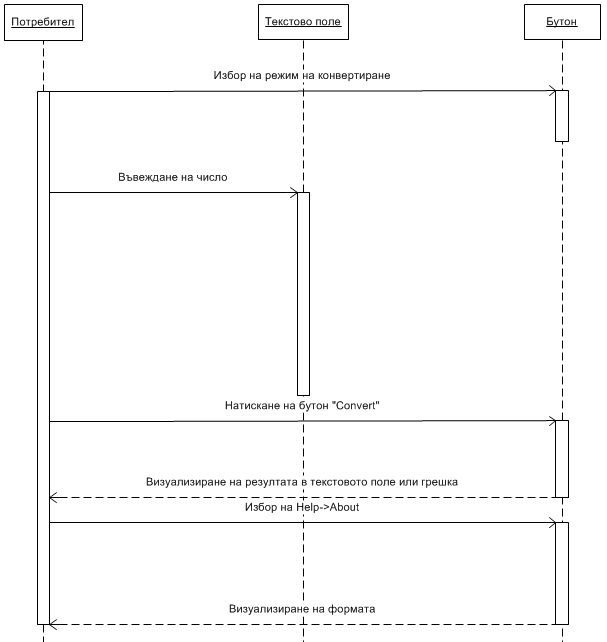




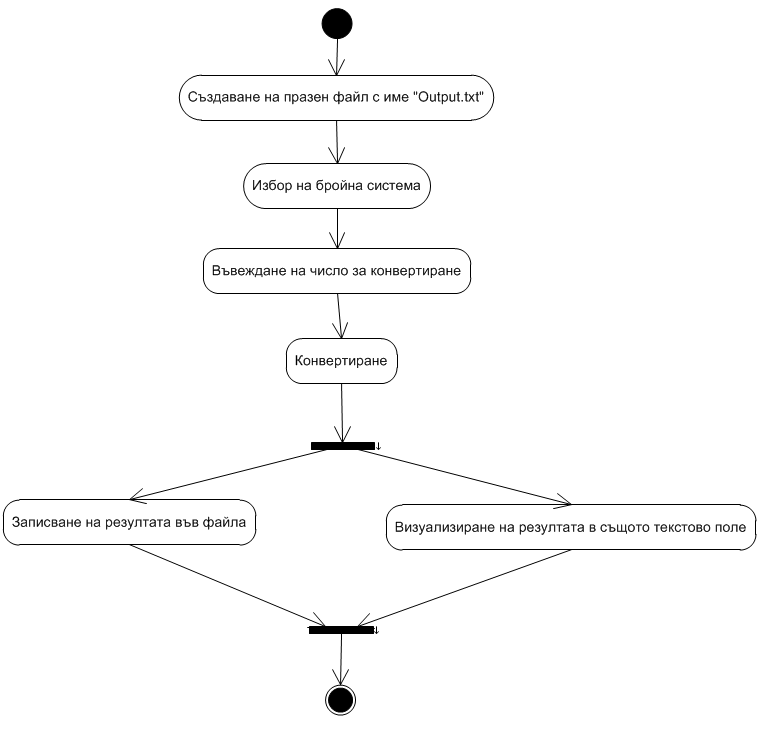
## Use Case



## Sequence



## Activity

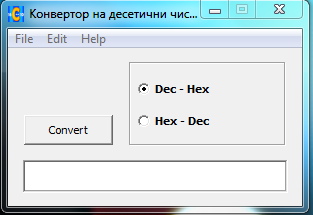


## Class



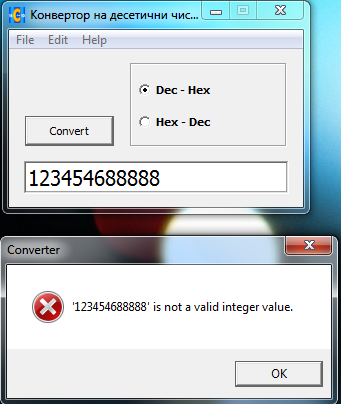
# Работа с програмата

След като се стартира програма на екрана се появява прозорецът от фиг. 1.



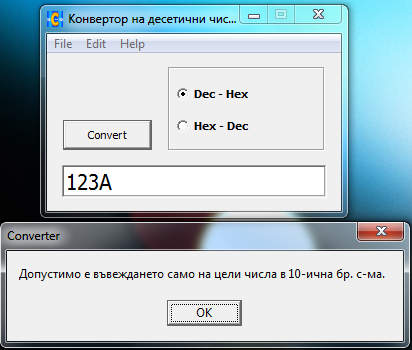
Фиг. Прозорец на програмата

С помощта на радио-бутоните Dec – Hex и Hex – Dec се избира от коя в коя бройна система да се конвертира. Dec – Hex е за конвертиране на цели числа от десетична бройна в шестнадесетична, а Hex – Dec служи обратната операция (от 16 в 10 бройна система). В текстовото поле се задава желаното число, като трябва да се има в предвид , че числото трябва да може да се съхранява в integer променлива, в противен случай се извежда съобщение (фиг. 2).



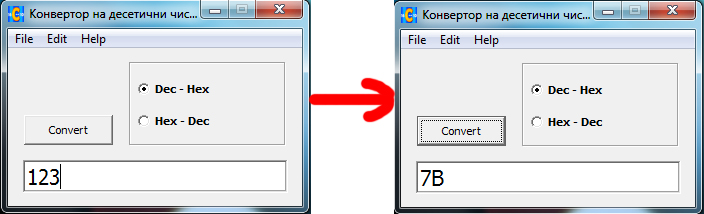
Фиг. Грешна стойност

Ако числото е в допустимите граници след избиране на бутона „Convert” се прави проверка дали числото е валидно. Ако не е, извежда се съобщение (фиг. 3).



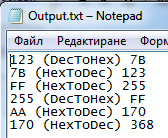
Фиг. Валидация

Но ако е валидно - резултатът се появява в същото текстово поле (фиг. 4).



Фиг. Извеждане на резултата

Резултатите автоматично се записват във файла „Output.txt“, която се създава в същата директория с програмата още със стартирането й (фиг. 5).



Фиг. Изходен файл

За информация относно програмата се избира Help->About.

За изход от програмата се избира File->Exit.

# Код на програмата

## KMain.pas

unit KMain;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ExtCtrls, Menus, Buttons;

type

TfrmMain = class(TForm)

editNumb: TEdit;

frmMainMenu: TMainMenu;

File1: TMenuItem;

Exit1: TMenuItem;

Edit2: TMenuItem;

Cut1: TMenuItem;

Copy1: TMenuItem;

Paste1: TMenuItem;

N1: TMenuItem;

Clear1: TMenuItem;

Help1: TMenuItem;

About1: TMenuItem;

btnConvert: TButton;

rdBtnGroupBox: TGroupBox;

rdBtnToHex: TRadioButton;

rdBtnToDec: TRadioButton;

frmPopupMenu: TPopupMenu;

Cut2: TMenuItem;

Copy2: TMenuItem;

Paste2: TMenuItem;

N2: TMenuItem;

Clear2: TMenuItem;

procedure editNumbKeyPress(Sender: TObject; var Key: Char);

procedure btnConvertClick(Sender: TObject);

procedure Exit1Click(Sender: TObject);

procedure Cut1Click(Sender: TObject);

procedure Copy1Click(Sender: TObject);

procedure Paste1Click(Sender: TObject);

procedure Clear1Click(Sender: TObject);

procedure About1Click(Sender: TObject);

procedure Cut2Click(Sender: TObject);

procedure Copy2Click(Sender: TObject);

procedure Paste2Click(Sender: TObject);

procedure Clear2Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

frmMain: TfrmMain;

txtFile: TextFile;

possNumbs: set of char = ['A'..'F', 'a'..'f', '0'..'9', #8];

procedure SaveToFile(Str: string);

function HexToDec(Str: string): integer;

function Verification(Str: string): boolean;

implementation

uses About;

{$R \*.dfm}

procedure SaveToFile(Str: string);

begin

{$I-} Append(txtFile); {$I+}

if ioresult <> 0 then begin

ShowMessage('Файлът не може да бъде намерен!');

exit;

end

else

WriteLn(txtFile, Str);

CloseFile(txtFile);

end;

function HexToDec(Str: string): integer;

var

i, M: integer;

begin

Result := 0;

M := 1;

Str := AnsiUpperCase(Str);

for i := Length(Str) downto 1 do begin

case Str[i] of

'1'..'9' :

Result := Result + (Ord(Str[i]) - Ord('0')) \* M;

'A'..'F' :

Result := Result + (Ord(Str[i]) - Ord('A') + 10) \* M;

end;

M := M shl 4;

end;

end;

{

Забранено е въвеждането на символи, освен разрешените (0..9 и A..F), но

други символи могат да бъдат въвеждани в текстово поле чрез "copy-paste".

Работата на тази функция е да провери дали има такива символи.

И ако конвертирането е 10->16 да провери дали всички символи са числа.

}

function Verification(Str: string): boolean;

var

i: integer;

setX: set of char;

begin

Result := true;

if (frmMain.rdBtnToDec.Checked) then setX := possNumbs

else setX := ['0'..'9'];

for i := 1 to Length(Str) do

if (Str[i] in setX = false) then begin

Result := false;

break;

end;

end;

procedure TfrmMain.editNumbKeyPress(Sender: TObject; var Key: Char);

begin

if Key = #13 then

btnConvertClick(Sender)

else if Key in possNumbs = false then

Key := #0;

end;

procedure TfrmMain.btnConvertClick(Sender: TObject);

var

msg: string;

begin

if (editNumb.Text <> '') then begin

if (rdBtnToHex.Checked) then begin

if (Verification(editNumb.Text)) then begin

msg := editNumb.Text;

editNumb.Text := Format('%0X', [StrToInt(editNumb.Text)]);

msg := msg + ' (DecToHex) ' + editNumb.Text;

SaveToFile(msg);

end

else

ShowMessage('Допустимо е въвеждането само на цели числа в 10-ична бр. с-ма.');

end

else begin

if (Verification(editNumb.Text)) then begin

msg := editNumb.Text;

editNumb.Text := IntToStr(HexToDec(editNumb.Text));

msg := msg + ' (HexToDec) ' + editNumb.Text;

SaveToFile(msg);

end

else

ShowMessage('Допустимо е въвеждането само на цели числа в 16-ична бр. с-ма.');

end;

end;

end;

procedure TfrmMain.Exit1Click(Sender: TObject);

begin

Close;

end;

procedure TfrmMain.Cut1Click(Sender: TObject);

begin

editNumb.CutToClipboard;

end;

procedure TfrmMain.Copy1Click(Sender: TObject);

begin

editNumb.CopyToClipboard;

end;

procedure TfrmMain.Paste1Click(Sender: TObject);

begin

editNumb.PasteFromClipboard;

end;

procedure TfrmMain.Clear1Click(Sender: TObject);

begin

editNumb.Clear;

end;

procedure TfrmMain.About1Click(Sender: TObject);

begin

frmAbout.Show;

end;

procedure TfrmMain.Cut2Click(Sender: TObject);

begin

Cut1Click(Sender);

end;

procedure TfrmMain.Copy2Click(Sender: TObject);

begin

Copy1Click(Sender);

end;

procedure TfrmMain.Paste2Click(Sender: TObject);

begin

Paste1Click(Sender);

end;

procedure TfrmMain.Clear2Click(Sender: TObject);

begin

Clear1Click(Sender);

end;

procedure TfrmMain.FormCreate(Sender: TObject);

begin

AssignFile(txtFile, 'Output.txt');

Rewrite(txtFile);

CloseFile(txtFile);

end;

end.

## About.pas

unit About;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls;

type

TfrmAbout = class(TForm)

lblUniName: TLabel;

lblAppName: TLabel;

btnClose: TButton;

procedure btnCloseClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

frmAbout: TfrmAbout;

implementation

{$R \*.dfm}

procedure TfrmAbout.btnCloseClick(Sender: TObject);

begin

Close;

end;

end.