**Задание:**

Разработать рекурсивную функцию нахождения значения функции Аккермана, которая определяется для всех неотрицательных целых аргументов m и n следующим образом:

A(0, n) = n + 1;

A(m, 0) = A(m – 1, 1);

A(m, n) = A(m – 1, A(m, n – 1);

**Код Delphi 10:**

**unit** Unit42;

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Menus;

**type**

TLab\_4\_2 = **class**(TForm)

MainMenu1: TMainMenu;

PopupMenu1: TPopupMenu;

SaveFile: TSaveDialog;

OpenFile: TOpenDialog;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

N: TEdit;

Calculate: TButton;

File1: TMenuItem;

Open: TMenuItem;

Save: TMenuItem;

N1: TMenuItem;

**Exit**: TMenuItem;

Help: TMenuItem;

Developer: TMenuItem;

Info: TMenuItem;

M: TEdit;

Result: TLabel;

Retry: TButton;

Answer: TEdit;

**procedure** CalculateClick(Sender: TObject);

**procedure** FormCreate(Sender: TObject);

**procedure** MChange(Sender: TObject);

**procedure** NChange(Sender: TObject);

**procedure** NKeyPress(Sender: TObject; **var** Key: Char);

**procedure** RetryClick(Sender: TObject);

**procedure** ExitClick(Sender: TObject);

**procedure** FormCloseQuery(Sender: TObject; **var** CanClose: Boolean);

**procedure** DeveloperClick(Sender: TObject);

**procedure** InfoClick(Sender: TObject);

**procedure** SaveClick(Sender: TObject);

**procedure** OpenClick(Sender: TObject);

**procedure** MKeyPress(Sender: TObject; **var** Key: Char);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

Lab\_4\_2: TLab\_4\_2;

**implementation**

{$R \*.dfm}

**function** Recur( NumM, NumN: Integer): Integer;

**var**

Num: Integer;

**begin**

**if** NumM = 0 **then**

Num := NumN + 1;

**if** (NumN = 0) **and** (NumM > 0) **then**

Num := Recur(NumM - 1, 1);

**if** (NumM > 0) **and** (NumN > 0) **then**

Num := Recur(NumM - 1, Recur(NumM , NumN - 1));

Recur := Num;

**end**;

**procedure** TLab\_4\_2.DeveloperClick(Sender: TObject);

**begin**

MessageDlg('Developer: Maksim Gladkiy.', mtInformation, [mbOk], 0);

**end**;

**procedure** TLab\_4\_2.InfoClick(Sender: TObject);

**begin**

MessageDlg('This program find''s the Akkerman''s function value with given

arguments'#13#10'using recursion according to the

conditions:'#13#10'A(0,n) = n + 1'#13#10'A(m,0) = A(m - 1,

1)'#13#10'A(m,n) = A(m - 1,A(m, n - 1))'#13#10'M is [0..3], N

is [0..9].', mtInformation, [mbOk], 0);

**end**;

**procedure** TLab\_4\_2.CalculateClick(Sender: TObject);

**var**

NumM, NumN, Ans: Integer;

**begin**

NumM := StrToInt(M.Text);

NumN := StrToInt(N.Text);

Ans := Recur(NumM, NumN);

M.Enabled := False;

N.Enabled := False;

Calculate.Enabled := False;

Result.Visible := True;

Answer.Visible := True;

Answer.Text := IntToStr(Ans);

Retry.Visible := True;

Save.Enabled := True;

**end**;

**procedure** TLab\_4\_2.ExitClick(Sender: TObject);

**begin**

Close;

**end**;

**procedure** TLab\_4\_2.FormCloseQuery(Sender: TObject; **var** CanClose: Boolean);

**var**

ButtonSelected: Byte;

**begin**

ButtonSelected := MessageDlg('Are you sure you want to exit?',

mtConfirmation, [mbYes,mbNo], 0);

**if** ButtonSelected <> mrYes **then**

CanClose := False;

**end**;

**procedure** TLab\_4\_2.FormCreate(Sender: TObject);

**begin**

Save.Enabled := False;

N.Enabled := False;

Calculate.Enabled := False;

Answer.Visible := False;

Retry.Visible := False;

Result.Enabled := False;

**end**;

**procedure** TLab\_4\_2.MChange(Sender: TObject);

**begin**

Save.Enabled := False;

**if** (Length(M.Text) > 0) **then**

N.Enabled := True

**else**

N.Enabled := False;

**end**;

**procedure** TLab\_4\_2.NKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** Char;

**begin**

Numerals := ['0'..'9', #8];

**with** Sender **as** TEdit **do**

**begin**

**if** (Key = #13) **and** (Length(Text) = 0) **then**

Key := #0;

**if not** (Key **in** Numerals) **then**

Key := #0;

**if** (Length(Text) = 1) **and** (Key <> #8) **then**

Key := #0;

**end**;

**end**;

**procedure** TLab\_4\_2.MKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** Char;

**begin**

Numerals := ['0'..'3', #8];

**with** Sender **as** TEdit **do**

**begin**

**if** (Key = #13) **and** (Length(Text) = 0) **then**

Key := #0;

**if not** (Key **in** Numerals) **then**

Key := #0;

**if** (Length(Text) = 1) **and** (Key <> #8) **then**

Key := #0;

**end**;

**end**;

**procedure** TLab\_4\_2.NChange(Sender: TObject);

**begin**

Save.Enabled := False;

**if** (Length(M.Text) > 0) **then**

Calculate.Enabled := True

**else**

Calculate.Enabled := False;

**end**;

**procedure** TLab\_4\_2.OpenClick(Sender: TObject);

**var**

InputFile: TextFile;

Temp: Double;

**begin**

M.Clear;

N.Clear;

**if** OpenFile.Execute **then**

**begin**

**try**

AssignFile(InputFile, OpenFile.FileName);

Reset(InputFile);

**if** SeekEof(InputFile) **then**

**begin**

MessageDlg('This file is empty. Try again.',

mtError, [mbRetry], 0);

CloseFile(InputFile);

**end**

**else**

**begin**

Read(InputFile, Temp);

M.Text := FloatToStr(Temp);

Readln(InputFile, Temp);

N.Text := FloatToStr(Temp);

CalculateClick(Sender);

CloseFile(InputFile);

**end**;

**except**

MessageDlg('Check entered data. Try again.', mtError, [mbRetry], 0);

M.Clear;

N.Clear;

CloseFile(InputFile);

**end**;

**end**;

**end**;

**function** CheckFileName(MyFile: String): String;

**var**

i: Byte;

IsCorrect: Boolean;

**begin**

IsCorrect := False;

i := 1;

**while not** IsCorrect **and** (i <= Length(MyFile)) **do**

**begin**

**if** MyFile[i] = '.' **then**

IsCorrect := True;

Inc(i);

**end**;

**if not** IsCorrect **then**

MyFile := MyFile + '.txt';

CheckFileName := MyFile;

**end**;

**procedure** TLab\_4\_2.SaveClick(Sender: TObject);

**var**

OutputFile: TextFile;

MyFile: String;

ButtonSelected, i, j: Byte;

**begin**

**if** SaveFile.Execute **then**

**begin**

MyFile := SaveFile.FileName;

MyFile := CheckFileName(MyFile);

**if** FileExists(MyFile) **then**

**begin**

ButtonSelected := MessageDlg('Do you want to rewrite the file?',

mtConfirmation, [mbYes,mbNo], 0);

**if** ButtonSelected = mrYes **then**

**begin**

AssignFile(OutputFile, MyFile);

Rewrite(OutputFile);

Writeln(OutputFile, Result.Caption,Answer.Text);

CloseFile(OutputFile);

**end**

**else**

**begin**

AssignFile(OutputFile, MyFile);

Append(OutputFile);

Writeln(OutputFile, Result.Caption,Answer.Text);

CloseFile(OutputFile);

**end**;

**end**

**else**

**begin**

AssignFile(OutputFile, MyFile);

Rewrite(OutputFile);

Writeln(OutputFile, Result.Caption,Answer.Text);

CloseFile(OutputFile);

**end**;

**end**;

**end**;

**procedure** TLab\_4\_2.RetryClick(Sender: TObject);

**begin**

M.Enabled := True;

Save.Enabled := False;

M.Clear;

N.Clear;

Answer.Clear;

Answer.Visible := False;

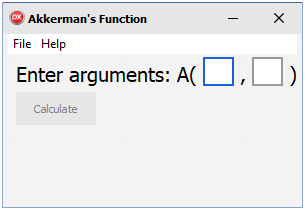
Result.Visible := False;

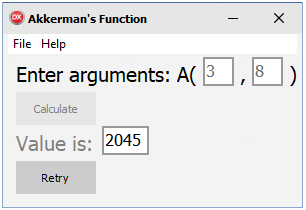
Retry.Visible := False;

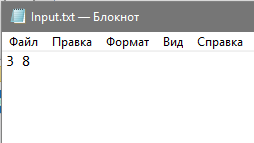
**end**;

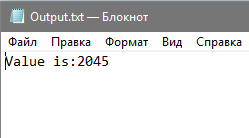
**end**.

**Скриншоты:**

****

****

****

****

**Блок-схема:**





