**УО «Белорусский государственный университет информатики и радиоэлектроники»**

**Кафедра ПОИТ**

**Отчёт по лабораторной работе №6.1**

**По предмету**

**Основы алгоритмизации и программирования**

**Вариант 6**

**Выполнил:**

**Гладкий М.Г.**

**Проверила:**

**Данилова Г.В.**

**Группа 851001**

**Минск 2019**

**Задание:**

Списки: Разработать программу работы с односвязным списком. Программа должна содержать следующие процедуры, вызываемые из меню:  
- построение пустого списка;  
- добавление нового элемента;  
- удаление указанного элемента;  
- подсчет количества всех вхождений заданного элемента;  
- просмотр списка.

**Delphi 10:**

**Main.pas**

**unit** Main;

**interface**

**uses**

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

Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Menus, Vcl.Grids,

Vcl.StdCtrls;

**type**

Pointer = ^TText;

TText = **Record**

Data: String[15];

Next: Pointer;

**end**;

TMainForm = **class**(TForm)

SG: TStringGrid;

MainMenu: TMainMenu;

PopupMenu: TPopupMenu;

About: TMenuItem;

Developer: TMenuItem;

Info: TMenuItem;

Add: TButton;

Delete: TButton;

Show: TButton;

Label1: TLabel;

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

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

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

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

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

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

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

**public**

Head, CopyHead, Buff: Pointer;

i: Byte;

**end**;

**var**

MainForm: TMainForm;

**implementation**

{$R \*.dfm}

**uses** AddText;

**procedure** TMainForm.AddClick(Sender: TObject);

**begin**

Visible := false;

**with** AddF **as** TAddF **do**

**begin**

TextT.Clear;

Submit.Enabled := false;

ShowModal;

**end**;

**end**;

**procedure** TMainForm.DeleteClick(Sender: TObject);

**var**

ButtonSelected, j, k: ShortInt;

Del, Prev: Pointer;

**begin**

**if** (SG.Row <> 0) **then**

**begin**

ButtonSelected := MessageDlg('Are you sure you want to delete this record?',

mtConfirmation, [mbYes,mbNo], 0);

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

**begin**

Del := CopyHead;

j := 0;

Head := CopyHead;

**while** (SG.Row - 1) <> j **do**

**begin**

Del := Del^.Next;

inc(j);

**end**;

**if** Del = CopyHead **then**

**begin**

Head := Del^.Next;

CopyHead := Head;

**for** k := 1 **to** i - 2 **do**

SG.Cells[0, k] := SG.Cells[0, k + 1];

SG.Cells[0, i - 1] := '';

**end**

**else**

**begin**

Prev := Head^.Next;

**while** (Prev^.Next <> Del)**and**(Prev^.Next <> nil) **do**

Prev := Prev^.Next;

Prev^.Next := Del^.Next;

**for** k := j + 1 **to** i - 2 **do**

SG.Cells[0, k] := SG.Cells[0, k + 1];

SG.Cells[0, i - 1] := '';

**end**;

Dispose(Del);

SG.RowCount := i - 1;

dec(i);

**end**;

**if** SG.RowCount = 1 **then**

Delete.Enabled := false;

**if** SG.RowCount > 2 **then**

Show.Enabled := true

**else**

Show.Enabled := false;

**end**;

**end**;

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

**begin**

i := 1;

Delete.Enabled := false;

Show.Enabled := false;

**with** SG **as** TStringGrid **do**

**begin**

RowCount := 1;

Cells[0, 0] := 'Text';

**end**;

**New**(Head);

Head^.Data := '';

Head^.Next := nil;

CopyHead := Head;

**end**;

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

**begin**

MessageDlg('This program is developed by Maksim Gladkiy, group 851001.',

mtInformation, [mbOk], 0);

**end**;

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

**begin**

MessageDlg('This program is Single Linked List.' + #10#13 + 'You can add and

delete any text in it.' + #10#13 + 'Also it shows number of meetings

of selected element.', mtInformation, [mbOk], 0);

**end**;

**procedure** TMainForm.ShowClick(Sender: TObject);

**var**

Str, Answ: String[15];

z: Integer;

**begin**

**if** (SG.Row <> 0) **then**

**begin**

z := 0;

Head := CopyHead;

Str := SG.Cells[0, SG.Row];

**while** Head <> nil **do**

**begin**

**if** Head^.Data = Str **then**

inc(z);

Head := Head^.Next;

**end**;

MessageDlg('This record meets ' + IntToStr(z) + ' times', mtInformation,

[mbOk], 0);

**end**;

**end**;

**end**.

**AddText.pas**

**unit** AddText;

**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**

TAddF = **class**(TForm)

Submit: TButton;

AddMenu: TMainMenu;

Help: TMenuItem;

Cancel: TButton;

TextT: TEdit;

Label1: TLabel;

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

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

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

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

**procedure** FormClose(Sender: TObject; **var** Action: TCloseAction);

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

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

AddF: TAddF;

**implementation**

{$R \*.dfm}

**uses** Main;

**procedure** TAddF.SubmitClick(Sender: TObject);

**begin**

**with** MainForm **as** TMainForm **do**

**begin**

Head := CopyHead;

**while** Head^.Next <> nil **do**

Head := Head^.Next;

**New**(Buff);

Buff^.Data := TextT.Text;

Buff^.Next := nil;

Head^.Next := Buff;

Head := Head^.Next;

SG.RowCount := i + 1;

SG.Cells[0, i] := TextT.Text;

inc(i);

**if** SG.RowCount > 2 **then**

Show.Enabled := true

**else**

Show.Enabled := false;

**end**;

MainForm.Delete.Enabled := true;

MainForm.Visible := true;

**end**;

**procedure** TAddF.HelpClick(Sender: TObject);

**begin**

MessageDlg('Here you can add text to the list.' + #10#13 + 'Text length is

[1..15].' + #10#13 + 'Use [''A..z'' ; ''-'' ; ''.'']', mtInformation, [mbOk], 0);

**end**;

**procedure** TAddF.FormClose(Sender: TObject; **var** Action: TCloseAction);

**begin**

MainForm.Visible := true;

**end**;

**procedure** TAddF.CancelClick(Sender: TObject);

**begin**

MainForm.Visible := true;

**end**;

**procedure** TAddF.PrChange(Sender: TObject);

**begin**

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

Submit.Enabled := true

**else**

Submit.Enabled := false;

**end**;

**procedure** TAddF.TextTKeyPress(Sender: TObject; **var** Key: Char);

**var**

Letters: **set of** Char;

**begin**

Letters := ['A'..'z', '.', '-', ' ', #8];

**with** TextT **as** TEdit **do**

**begin**

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

Key := #0;

**if** (Length(Text) <> SelStart) **then**

Key := #0;

**if** (Length(Text) = 15) **and** (Key <> #8)**then**

Key := #0

**end**;

**end**;

**end**.

**library ListLib**

**uses**

System.SysUtils,

System.Classes;

**type**

Pointer = ^TText;

TText = Record

Data: String;

Next: Pointer;

end;

{$R \*.res}

**procedure** \_AddHead(**var** Head, CopyHead: Pointer); stdcall;

**begin**

New(Head);

Head^.Data := '';

Head^.Next := nil;

CopyHead := Head;

**end;**

**procedure** \_AddEl(**var** Head, Buff: Pointer; **var** Str: String); stdcall;

**begin**

**while** Head^.Next <> nil **do**

Head := Head^.Next;

New(Buff);

Buff^.Data := Str;

Buff^.Next := nil;

Head^.Next := Buff;

Head := Head^.Next;

**end**;

**exports**

\_AddHead,

\_AddEl;

**begin**

**end**.

**unit MyUnit**

**interface**

**uses**

Vcl.Grids, Vcl.Dialogs, System.SysUtils;

**type**

Pointer = ^TText;

TText = Record

Data: String;

Next: Pointer;

**end**;

procedure HeadCreate;

procedure DeliteEl(SG: TStringGrid);

procedure AddEl(St: String);

procedure ShowAns(SG: TStringGrid);

**var**

Head, CopyHead, Buff: Pointer;

i: Byte;

**implementation**

**procedure** HeadCreate;

**begin**

i := 1;

New(Head);

Head^.Data := '';

Head^.Next := nil;

CopyHead := Head;

**end**;

**procedure** DeliteEl(SG: TStringGrid);

**var**

j, k: ShortInt;

Del, Prev: Pointer;

**begin**

Del := CopyHead;

j := 0;

Head := CopyHead;

**while** (SG.Row - 1) <> j **do**

**begin**

Del := Del^.Next;

inc(j);

**end**;

**if** Del = CopyHead **then**

**begin**

Head := Del^.Next;

CopyHead := Head;

**for** k := 1 **to** i - 2 **do**

SG.Cells[0, k] := SG.Cells[0, k + 1];

SG.Cells[0, i - 1] := '';

**end**

**else**

**begin**

Prev := Head^.Next;

**while** (Prev^.Next <> Del)**and**(Prev^.Next <> nil) **do**

Prev := Prev^.Next;

Prev^.Next := Del^.Next;

**for** k := j + 1 **to** i - 2 **do**

SG.Cells[0, k] := SG.Cells[0, k + 1];

SG.Cells[0, i - 1] := '';

**end**;

Dispose(Del);

SG.RowCount := i - 1;

dec(i);

**end**;

**procedure** AddEl(St: String);

**begin**

Head := CopyHead;

**while** Head^.Next <> nil **do**

Head := Head^.Next;

New(Buff);

Buff^.Data := St;

Buff^.Next := nil;

Head^.Next := Buff;

Head := Head^.Next;

**end**;

**procedure** ShowAns(SG: TStringGrid);

**var**

z: ShortInt;

Str: String;

**begin**

z := 0;

Head := CopyHead;

Str := SG.Cells[0, SG.Row];

**while** Head <> nil **do**

**begin**

**if** Head^.Data = Str **then**

inc(z);

Head := Head^.Next;

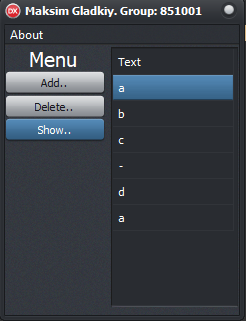
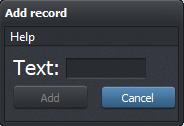
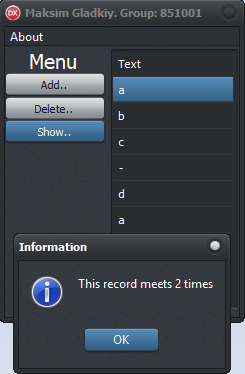
**end**;

MessageDlg('This record meets ' + IntToStr(z) + ' times', mtInformation, [mbOk],

0);

**end**;

**end**.

**Скриншоты:  
  
  
**

**Блок-схема:**





