**УО «Белорусский государственный университет информатики и**

**радиоэлектроники»**

**Кафедра ПОИТ**

**Отчет по лабораторной работе №4.2**

**по предмету**

**Основы Алгоритмизации и Программирования**

**Вариант 1**

**Выполнил**

**Андросов И.С.**

**Проверила**

**Данилова Г.В.**

Группа:

**8**51001

**Минск 2019**

**Задание**

**Рекурсия**

Разработать рекурсивную процедуру двоичного поиска элемента массива, равного данному числу.

**Код программы**

**(Delphi)**

**unit** Lab4\_2;

**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, Vcl.Samples.Spin;

**type**

TArray = **array of** SmallInt;

TIlya = **class**(TForm)

MainMenu: TMainMenu;

SG: TStringGrid;

PopupMenu: TPopupMenu;

SaveFile: TSaveDialog;

OpenFile: TOpenDialog;

FileMenu: TMenuItem;

HelpMenu: TMenuItem;

Open: TMenuItem;

Save: TMenuItem;

N1: TMenuItem;

**Exit**: TMenuItem;

AboutTheProgram: TMenuItem;

AboutTheDeveloper: TMenuItem;

EnterSize: TLabel;

SetSize: TButton;

SizeEdit: TSpinEdit;

Answer: TLabel;

EnterNumber: TLabel;

NumberEdit: TEdit;

FindNum: TButton;

**procedure** CheckEmtyCells(**var** MyArray: TArray; Sender: TObject);

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

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

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

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

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

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

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

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

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

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

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

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

**procedure** SGKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**procedure** NumberEditKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**procedure** SizeEditKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**end**;

**var**

Ilya: TIlya;

**implementation**

{$R \*.dfm}

**function** BinarySearch(MyArray: TArray; High, Low: Byte; Number: SmallInt): ShortInt;

**var**

Mid: Byte;

**begin**

Mid := (High + Low) **div** 2;

**if** (Number = MyArray[Mid]) **then**

BinarySearch := Mid

**else**

**if** High > Low **then**

**if** (Number < MyArray[Mid]) **then**

BinarySearch := BinarySearch(MyArray, Mid - 1, Low, Number)

**else**

BinarySearch := BinarySearch(MyArray, High, Mid + 1, Number)

**else**

BinarySearch := -1;

**end**;

**function** OrderArray(MyArray: TArray): TArray;

**var**

ThatArray: TArray;

i, j, Left, Right, Size: Byte;

Temp: SmallInt;

**begin**

Size := High(MyArray);

Left := Size;

Right := Size;

SetLength(ThatArray, Size \* 2 + 1);

ThatArray[Size] := MyArray[0];

**for** i := 1 **to** Size **do**

**begin**

Temp := MyArray[i];

**if** Temp >= MyArray[0] **then**

**begin**

Inc(Right);

j := Right;

**while** Temp < ThatArray[j - 1] **do**

**begin**

ThatArray[j] := ThatArray[j-1];

Dec(j);

**end**;

ThatArray[j] := Temp;

**end**

**else**

**begin**

Dec(Left);

j := Left;

**while** Temp > ThatArray[j + 1] **do**

**begin**

ThatArray[j] := ThatArray[j + 1];

Inc(j);

**end**;

**end**;

ThatArray[j] := Temp;

**end**;

**for** j := 0 **to** Size **do**

MyArray[j] := ThatArray[j + Left];

OrderArray := MyArray;

**end**;

**function** CheckArray(MyArray: TArray): Boolean;

**var**

IsIncreasing: Boolean;

i: Byte;

**begin**

IsIncreasing := true;

**for** i := 0 **to** High(MyArray) - 1 **do**

**if** (IsIncreasing) **and** (MyArray[i] > MyArray[i + 1]) **then**

IsIncreasing := false;

CheckArray := IsIncreasing;

**end**;

**procedure** TIlya.CheckEmtyCells(**var** MyArray: TArray; Sender: TObject);

**const**

Error = 'Empty cells were filled with zeros.';

**var**

Empty: Boolean;

i, Size: Byte;

**begin**

Empty := false;

Size := StrToInt(SizeEdit.Text);

SetLength(MyArray, Size);

**for** i := 0 **to** High(MyArray) **do**

**begin**

**if** (SG.Cells[i,1] = '') **or** (SG.Cells[i,1] = '-') **then**

**begin**

Empty := true;

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

**end**;

MyArray[i] := StrToInt(SG.Cells[i, 1]);

**end**;

**if** Empty **then**

MessageDlg(Error, mtConfirmation, [mbOk], 0);

**end**;

**procedure** TIlya.FindNumClick(Sender: TObject);

**const**

Hint = 'The array has been ordered.';

**var**

MyArray: TArray;

i, Index: ShortInt;

**begin**

Save.Enabled := true;

CheckEmtyCells(MyArray, Sender);

**if** (**not**(CheckArray(MyArray))) **then**

**begin**

MyArray := OrderArray(MyArray);

MessageDlg(Hint, mtConfirmation, [mbOk], 0);

**for** i := 0 **to** High(MyArray) **do**

SG.Cells[i, 1] := IntToStr(MyArray[i])

**end**;

Index := BinarySearch(MyArray, High(MyArray), 0, StrToInt(NumberEdit.Text));

**if** Index <> -1 **then**

Answer.Caption := 'Number ' + NumberEdit.Text + ' stands in array by the index '

+ IntToStr(Index + 1)

**else**

Answer.Caption := 'Number ' + NumberEdit.Text + ' is not in array.';

**end**;

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

**var**

InputFile: TextFile;

i, j: Byte;

Temp: SmallInt;

**begin**

**for** j := 0 **to** 1 **do**

**for** i := 0 **to** 14 **do**

SG.Cells[i, j] := '';

NumberEdit.Clear;

Answer.Caption := '';

**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**

ReadLn(InputFile, Temp);

**if** (Temp > 15) **or** (Temp < 2) **then**

MessageDlg('Check entered data. Try again.', mtError,

[mbRetry], 0)

**else**

**begin**

SizeEdit.Text := IntToStr(Temp);

SetSizeClick(Sender);

i := 0;

**repeat**

Read(InputFile, Temp);

SG.Cells[i, 1] := IntToStr(Temp);

Inc(i);

**until** EoLn(InputFile);

ReadLn(InputFile);

Read(InputFile, Temp);

NumberEdit.Text := IntToStr(Temp);

FindNumClick(Sender);

**end**;

CloseFile(InputFile);

**end**;

**except**

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

CloseFile(InputFile);

**end**;

**end**;

**end**;

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

**var**

OutputFile: TextFile;

MyFile: String;

ButtonSelected: Byte;

**begin**

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

**begin**

MyFile := SaveFile.FileName;

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

**begin**

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

mtConfirmation, [mbYes,mbNo], 0);

AssignFile(OutputFile, MyFile);

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

Rewrite(OutputFile)

**else**

**begin**

Append(OutputFile);

WriteLn(OutputFile);

**end**;

Write(OutputFile, Answer.Caption);

CloseFile(outputFile);

**end**;

**end**;

**end**;

**procedure** TIlya.SizeEditChange(Sender: TObject);

**var**

i, j: Byte;

**begin**

NumberEdit.Text := '';

NumberEdit.Enabled := false;

Save.Enabled := false;

Answer.Caption := '';

FindNum.Enabled := false;

SG.Enabled := false;

**for** j := 0 **to** 1 **do**

**for** i := 0 **to** 14 **do**

Sg.Cells[i, j] := '';

**end**;

**procedure** TIlya.SizeEditKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**begin**

**if** SizeEdit.Text = '' **then**

SizeEdit.Text := '2';

**end**;

**procedure** TIlya.SetSizeClick(Sender: TObject);

**var**

i: Byte;

**begin**

NumberEdit.Enabled := true;

SG.Enabled := true;

SG.ColCount := StrToInt(SizeEdit.Text);

**for** i := 1 **to** SG.ColCount **do**

SG.Cells[i - 1, 0] := IntToStr(i);

**end**;

**procedure** TIlya.SGKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **Set of** char;

**begin**

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

Save.Enabled := false;

Answer.Caption := '';

**if** NumberEdit.Text <> '' **then**

FindNum.Enabled := true

**else**

FindNum.Enabled := false;

**with** Sender **as** TStringGrid **do**

**begin**

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

Key := #0;

**if** (Length(Cells[Col, Row]) = 3) **and** (Key <> #8) **then**

Key := #0;

**if** (Cells[Col, Row] = '0') **and** (Key <> #8) **then**

Key := #0;

**if** (Length(Cells[Col, Row]) > 0) **and** (Key = #45) **then**

Key := #0;

**if** (Cells[Col, Row] = '-') **and** ((Key = '0') **or** (Key = '-')) **then**

Key := #0;

**if** (Length(Cells[Col, Row]) > 0) **and** (Key = '-') **then**

Key := #0;

**end**;

**end**;

**procedure** TIlya.SGKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**var**

Temp: SmallInt;

**begin**

**with** Sender **as** TStringGrid **do**

**begin**

**if** Length(Cells[Col, Row]) > 1 **then**

**try**

Temp := StrToInt(Cells[Col, Row]);

**except**

Cells[Col, Row] := '0';

**end**;

**if** (Cells[Col, Row] = '-0') **or** (Cells[Col, Row] = '00') **then**

Cells[Col, Row] := '0'

**end**;

**end**;

**procedure** TIlya.NumberEditKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **Set of** char;

**begin**

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

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

**begin**

**if** (FindNum.Enabled) **and** (Key = #13) **then**

FindNumClick(Sender);

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

Key := #0;

**if** (Length(Text) = 3) **and** (Key <> #8) **then**

Key := #0;

**if** (Text = '0') **and** (Key <> #8) **then**

Key := #0;

**if** (Length(Text) > 0) **and** (Key = #45) **then**

Key := #0;

**if** (Text = '-') **and** ((Key = '0') **or** (Key = '-')) **then**

Key := #0;

**if** (Length(Text) > 0) **and** (Key = '-') **then**

Key := #0;

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

Key := #0;

**end**;

**end**;

**procedure** TIlya.NumberEditKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**var**

Temp: SmallInt;

**begin**

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

**begin**

**if** Length(Text) > 1 **then**

**try**

Temp := StrToInt(Text);

**except**

Text := '0';

**end**;

**if** (Text = '-0') **or** (Text = '00') **then**

Text := '0'

**end**;

**end**;

**procedure** TIlya.NumberEditChange(Sender: TObject);

**begin**

**if** (Length(NumberEdit.Text) <> 0) **and** (NumberEdit.Text <> '-') **then**

FindNum.Enabled := true

**else**

FindNum.Enabled := false;

Save.Enabled := false;

Answer.Caption := '';

**end**;

**procedure** TIlya.AboutTheProgramClick(Sender: TObject);

**begin**

MessageDlg('This program searches for the number in a sorted array with a help of

binary search.' + #13 + 'Size should be in range 2..15, Numbers should be

in range -99..999' + #13 + 'Ender data in file as you would enter it in

this program.', mtInformation, [mbOk], 0);

**end**;

**procedure** TIlya.AboutTheDeveloperClick(Sender: TObject);

**begin**

MessageDlg('This program is developed by Ilya Androsov' + #13 + 'BSUIR 2019',

mtInformation, [mbOk], 0);

**end**;

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

**begin**

Close;

**end**;

**procedure** TIlya.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**

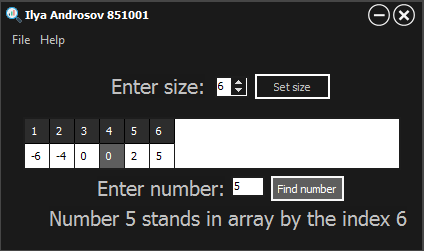
CanClose := False;

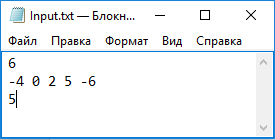
**end**;

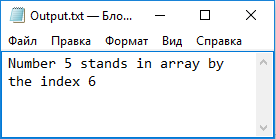
**end**.

**Скриншоты**

**(Delphi)**

****

****

****

**Схема алгоритма**





