УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №6.2

по предмету «Основы алгоритмизации и программирования»

Вариант 18

Выполнил:

Машевский Д.В

Студент гр. 351003

Проверил:

Данилова Г. В.

Минск 2024

**Код программы Delphi:**

**unit Unit4Laba62;**

interface

uses

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

System.Classes, Vcl.Graphics,

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

Vcl.Samples.Spin, Vcl.ExtCtrls, Clipbrd;

Const

COUNT\_OF\_NUMBERS = 1;

MAX\_LENGTH\_VALUE = 20;

type

TIntArr = Array of Integer;

TMainForm = class(TForm)

MainMenu: TMainMenu;

SpinEditEnterSize: TSpinEdit;

LabelGiveInfo: TLabel;

ButtonEnterSize: TButton;

MemoGiveResult: TMemo;

MemoOutPutSteps: TMemo;

PopupMenu: TPopupMenu;

NFile: TMenuItem;

NInstruction: TMenuItem;

NDeveloper: TMenuItem;

NFileOpen: TMenuItem;

NFileSave: TMenuItem;

NExitProgram: TMenuItem;

procedure ButtonEnterSizeClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

Function CountWays(Size: Integer; Steps: string): Integer;

procedure NExitProgramClick(Sender: TObject);

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

procedure NFileOpenClick(Sender: TObject);

procedure NFileSaveClick(Sender: TObject);

procedure SpinEditEnterSizeClick(Sender: TObject);

procedure SpinEditEnterSizeKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

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

procedure NInstructionClick(Sender: TObject);

procedure NDeveloperClick(Sender: TObject);

private

public

end;

var

MainForm: TMainForm;

implementation

{$R \*.dfm}

Function TMainForm.CountWays(Size: Integer; Steps: string): Integer;

Begin

If Size = 0 Then

Begin

MemoOutPutSteps.Lines.Add(Steps);

Sleep(0);

CountWays := 1;

End

Else If Size > 0 Then

Begin

CountWays := CountWays(Size - 1, Steps + '1') +

CountWays(Size - 2, Steps + '2');

End

Else

Begin

CountWays := 0;

End;

End;

procedure TMainForm.ButtonEnterSizeClick(Sender: TObject);

var

Size: Integer;

begin

Size := SpinEditEnterSize.Value;

MemoOutPutSteps.Clear;

MemoOutPutSteps.Enabled := True;

MemoGiveResult.Clear;

MemoGiveResult.Text := IntToStr(CountWays(Size, ''));

MainMenu.Items[0].Items[1].Enabled := True;

end;

procedure TMainForm.FormCloseQuery(Sender: TObject; var CanClose: Boolean);

var

UserChoice: Integer;

Begin

CanClose := False;

UserChoice := MessageBox(Handle, 'Вы уверены, что хотите выйти?',

'Подтверждение', MB\_ICONQUESTION or MB\_YESNO);

If UserChoice = IDYES then

CanClose := True;

end;

procedure TMainForm.FormCreate(Sender: TObject);

begin

MemoGiveResult.Clear;

MemoGiveResult.Enabled := False;

MemoOutPutSteps.Clear;

MemoOutPutSteps.Enabled := False;

MainMenu.Items[0].Items[1].Enabled := False;

end;

Function CheckFileCount(FileInputPath: String): Boolean;

Var

InputFile: TextFile;

Value: Char;

IsDigit: Boolean;

IsCorrect: Boolean;

IsOkay: Boolean;

Counter: Integer;

Begin

Counter := 0;

AssignFile(InputFile, FileInputPath);

Reset(InputFile);

IsDigit := True;

IsCorrect := True;

While not(Eof(InputFile)) do

Begin

Try

Read(InputFile, Value);

Except

IsCorrect := False;

End;

If (Value = ' ') and (IsDigit = True) then

IsDigit := False;

If (Value <> ' ') and (IsDigit = False) then

Begin

Inc(Counter);

IsDigit := True;

End;

End;

Inc(Counter);

If (Counter <> COUNT\_OF\_NUMBERS) or (not IsCorrect) then

Begin

ShowMessage

('В файле некорректные данные:'#10'В файле должна содержаться 1 переменная целого типа');

CheckFileCount := False;

End

Else

CheckFileCount := True;

End;

Function CheckFile(Path: String): Boolean;

Var

Fl: Boolean;

InputFile: TextFile;

Begin

Fl := True;

Try

AssignFile(InputFile, Path);

Reset(InputFile);

Except

ShowMessage('Произошла ошибка:файл занят другим процессом');

Fl := False;

End;

CheckFile := Fl;

CloseFile(InputFile);

End;

Procedure ReadSet(FileName: String; var D: TIntArr);

Var

InputFile: TextFile;

IsSymbol, IsCorrect: Boolean;

I: Integer;

Begin

AssignFile(InputFile, FileName);

Reset(InputFile);

SetLength(D, COUNT\_OF\_NUMBERS);

I := 0;

While not Eof(InputFile) and (I < COUNT\_OF\_NUMBERS) do

Begin

Try

IsCorrect := True;

Read(InputFile, D[I]);

Except

IsCorrect := False;

End;

Inc(I);

End;

CloseFile(InputFile);

End;

procedure TMainForm.NDeveloperClick(Sender: TObject);

Var

MessageBoxCaption: String;

MessageBoxText: String;

IsMessageShow: Boolean;

Begin

IsMessageShow := False;

if not IsMessageShow then

Begin

IsMessageShow := True;

MessageBoxCaption := 'О разработчике';

MessageBoxText :=

'Машевский Даниил Витальевич, группа 351003, Лабораторная

№6.2.'#10''#10'Задача:'#10'Имеется полоска клетчатой бумаги шириной в одну

клетку и длиной в n клеток. Разработать рекурсивную функцию, определяющую

количество способов продвижения шашки на n-ю клетку.';

MessageBox(Handle, PChar(MessageBoxText),

PChar(MessageBoxCaption), MB\_OK);

End;

IsMessageShow := False;

end;

procedure TMainForm.NExitProgramClick(Sender: TObject);

begin

MainForm.close;

end;

Function ReadSetCorrect(FileName: String; var D: TIntArr): Boolean;

Var

InputFile: TextFile;

IsSymbol, IsCorrect: Boolean;

I: Integer;

Begin

AssignFile(InputFile, FileName);

Reset(InputFile);

SetLength(D, COUNT\_OF\_NUMBERS);

I := 0;

While not Eof(InputFile) and (I < COUNT\_OF\_NUMBERS) do

Begin

Try

IsCorrect := True;

Read(InputFile, D[I]);

Except

IsCorrect := False;

End;

if D[I] < 0 then

Begin

IsCorrect := False;

End;

Inc(I);

End;

CloseFile(InputFile);

ReadSetCorrect := IsCorrect;

End;

procedure TMainForm.NFileOpenClick(Sender: TObject);

Var

OpenDialog: TOpenDialog;

InputFile: TextFile;

N, I: Integer;

D: TIntArr;

IsCorrect: Boolean;

Begin

IsCorrect := True;

OpenDialog := TOpenDialog.Create(nil);

OpenDialog.Title := 'Выберите файл';

OpenDialog.Filter := 'Текстовые файлы (\*.txt)|\*.txt';

If OpenDialog.Execute then

Begin

if CheckFile(OpenDialog.FileName) then

If (CheckFileCount(OpenDialog.FileName)) then

Begin

MainMenu.Items[0].Items[1].Enabled := True;

ReadSet(OpenDialog.FileName, D);

if D[0] > MAX\_LENGTH\_VALUE then

Begin

ShowMessage('Значения в файле не должны превышать 20');

End;

if ReadSetCorrect(OpenDialog.FileName, D) = False then

Begin

ShowMessage('Некорректные значения в файле');

End

Else

Begin

SpinEditEnterSize.Value := D[0];

End;

End;

End;

end;

procedure WriteMemoIntoFile(Path: string; Memo: TMemo);

var

F: TextFile;

I: Integer;

FResult: String;

begin

FResult := 'Вычисленное значение: ';

AssignFile(F, Path);

Rewrite(F);

Writeln(F, FResult);

for I := 0 to Memo.Lines.Count - 1 do

Writeln(F, Memo.Lines[I]);

CloseFile(F);

end;

procedure TMainForm.NFileSaveClick(Sender: TObject);

var

SaveDialog: TSaveDialog;

FileName: string;

MessageBoxCaption: String;

MessageBoxText: String;

IsMessageShow: Boolean;

begin

SaveDialog := TSaveDialog.Create(nil);

SaveDialog.Title := 'Сохранить файл';

SaveDialog.Filter := 'Текстовые файлы (\*.txt)|\*.txt';

SaveDialog.DefaultExt := 'txt';

if SaveDialog.Execute then

begin

FileName := SaveDialog.FileName;

if not FileName.EndsWith('.txt') then

FileName := FileName + '.txt';

WriteMemoIntoFile(FileName, MemoGiveResult);

// Replace Memo1 with the name of your TMemo component

end;

SaveDialog.Free;

IsMessageShow := False;

if not IsMessageShow then

Begin

IsMessageShow := True;

MessageBoxCaption := 'Работа с файлом:';

MessageBoxText := 'Запись в файл прошла успешно!';

MessageBox(Handle, PChar(MessageBoxText),

PChar(MessageBoxCaption), MB\_OK);

End;

IsMessageShow := False;

end;

procedure TMainForm.NInstructionClick(Sender: TObject);

Var

MessageBoxCaption: String;

MessageBoxText: String;

IsMessageShow: Boolean;

Begin

IsMessageShow := False;

if not IsMessageShow then

Begin

IsMessageShow := True;

MessageBoxCaption := 'Инструкция';

MessageBoxText :=

'1) Вводимое значение не должно превышать 20.'#10'2) При работе с файлом,

файл должен быть формата .txt'#10'3) В файле должны быть записаны только

одно число <=20';

MessageBox(Handle, PChar(MessageBoxText),

PChar(MessageBoxCaption), MB\_OK);

End;

IsMessageShow := False;

end;

procedure TMainForm.SpinEditEnterSizeClick(Sender: TObject);

begin

Clipboard.Clear;

end;

procedure TMainForm.SpinEditEnterSizeKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if (Key = VK\_INSERT) and (SsShift in Shift) then

Key := 0;

if (Key = Ord('V')) and (SsCtrl in Shift) then

Key := 0;

end;

procedure TMainForm.SpinEditEnterSizeKeyPress(Sender: TObject; var Key: Char);

var

Value: Integer;

begin

if TryStrToInt(SpinEditEnterSize.Text + Key, Value) then

if Value > MAX\_LENGTH\_VALUE then

Key := #0;

end;

end.

**Код программы Java:**

import java.util.Scanner;

import java.io.\*;

public class Main {

static Scanner scanner = new Scanner(System.in);

public static int inputChoice() {

int vib;

boolean isIncorrect;

vib = 0;

System.out.println("Введите 0, если хотите сделать ввод с консоли; 1, если

хотите ввод из файла. ");

do {

isIncorrect = false;

System.out.print("Выбор: ");

try {

vib = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("Ошибка ввода. ");

}

if (!isIncorrect && (vib != 1) && (vib != 0)) {

isIncorrect = true;

System.out.println("Число должно быть либо 0, либо 1. ");

}

} while (isIncorrect);

return vib;

}

public static int enterSize() {

int size;

boolean isIncorrect;

size = 0;

do {

isIncorrect = false;

System.out.print("Введите размер поля: ");

try {

size = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("Ошибка ввода. ");

}

if (!isIncorrect && (size < 2)) {

isIncorrect = true;

System.out.println("Введено недопустимое число элементов. Повторите

ввод. ");

}

} while (isIncorrect);

return size;

}

public static String pTF() {

String path;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.println("Укажите путь к файлу: ");

path = scanner.nextLine();

File file = new File(path);

if (!file.exists()) {

isIncorrect = true;

System.out.println("Такой файл не найден. ");

}

if (!path.endsWith(".txt")) {

isIncorrect = true;

System.out.println("У файла должно быть расширение txt. ");

}

} while (isIncorrect);

return path;

}

public static int readNumberFile(String path) {

int n;

boolean isIncorrect;

n = 0;

isIncorrect = false;

try {

Scanner scannerFile = new Scanner(new File(path));

n = scannerFile.nextInt();

scannerFile.close();

} catch (Exception e) {

isIncorrect = true;

System.out.println("Ошибка считывания размера поля с файла. ");

}

if (!isIncorrect && n < 2) {

isIncorrect = true;

System.out.println("Значение размера поля в файле должно быть больше 1.

");

}

if (isIncorrect) {

System.out.println("Введите размер поля с консоли. ");

n = enterSize();

}

return n;

}

public static int outputChoice() {

int vib;

boolean isIncorrect;

vib = 0;

System.out.println("Введите 0, если хотите сделать вывод в консоль; 1, если

хотите вывод в файл. ");

do {

isIncorrect = false;

System.out.print("Выбор: ");

try {

vib = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("\nОшибка ввода. ");

}

if (!isIncorrect && (vib != 1) && (vib != 0)) {

isIncorrect = true;

System.out.println("Число должно быть либо 0, либо 1. ");

}

} while (isIncorrect);

return vib;

}

public static int countWays(int size) {

if (size == 1) {

return 1;

} else if (size == 2) {

return 2;

} else {

return countWays(size - 1) + countWays(size - 2);

}

}

public static void printResultNumber(int number) {

System.out.println(number);

}

public static void printResultNumberFile(int number, String path) {

try (PrintWriter writer = new PrintWriter(new FileWriter(path))) {

writer.print(number); // Use print() instead of write()

} catch (Exception e) {

System.out.println("Не удалось записать данные в файл.");

}

}

public static void getInfo() {

System.out.println("Данная программа определяет количество способов

продвижения шашки на n-ю клетку.");

}

public static void printFinal(int vib, int number) {

String path;

if (vib == 0) {

printResultNumber(number);

}

else {

path = pTF();

printResultNumberFile(number, path);

System.out.println("Запись в файл прошла успешно!");

}

}

public static int madeChoice(int vib) {

String path;

int size;

if (vib == 0) {

size = enterSize();

} else {

path = pTF();

size = readNumberFile(path);

}

return size;

}

public static void main(String[] args) {

int vibInPut;

int vibOutPut;

int size;

int fNumber;

getInfo();

vibInPut = inputChoice();

size = madeChoice(vibInPut);

fNumber = countWays(size);

vibOutPut = outputChoice();

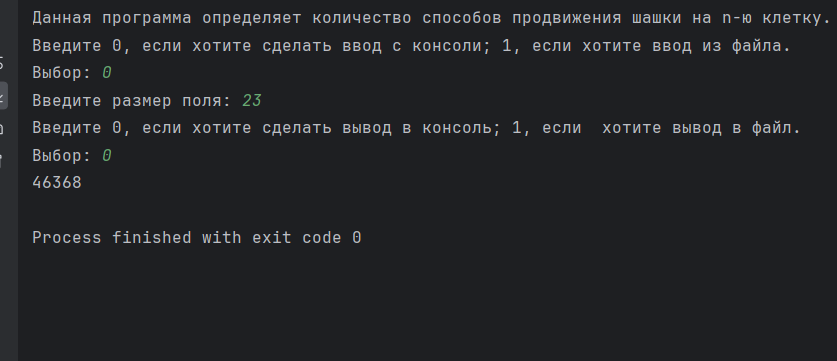
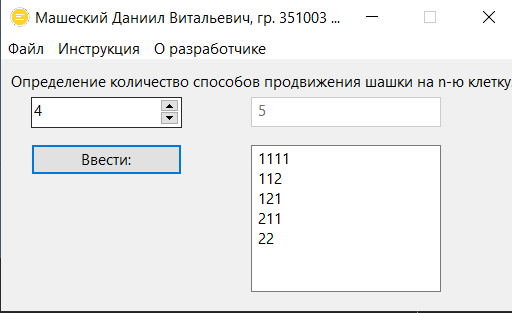
printFinal(vibOutPut, fNumber);

scanner.close();

}

}

**Скриншоты:**



**Блок-схема:**

