Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Глазовский государственный педагогический институт имени В.Г. Короленко»

**ОТЧЕТ**

**по учебной практике**

**ПМ 03 Участие в интеграции программных модулей**

**Специальность 09.02.03 Программирование в компьютерных системах**

|  |
| --- |
| Выполнил(а): |
| студент(ка) ОП СПО группа 43 |
|  |
|  |
| Руководитель практики: старший преподаватель кафедры математики и информатики Касаткин К.А. |

г. Глазов 2020

**Оглавление**

[Введение 3](#_Toc35594410)

[Обзор аналогов 4](#_Toc35594411)

[Постановка задачи и требования, предъявляемые к программе 6](#_Toc35594412)

[Описание алгоритма работы программы 7](#_Toc35594413)

[Выбор и обоснование выбора средства разработки, исходный код программы 8](#_Toc35594414)

[Тестирование программы 21](#_Toc35594415)

[Руководство по установке 22](#_Toc35594416)

[Руководство пользователя 23](#_Toc35594417)

[Заключение 24](#_Toc35594418)

[Список литературы 25](#_Toc35594419)

# **Введение**

На время прохождения учебной практики была выбрана тема – создание игры головоломки-оборотня “Бокалы”.

Игры оборотни – фишки для таких игр окрашены в два цвета (например, с одной стороны они белые, с другой черные) и время от времени они изменяют свой цвет на противоположный (иначе говоря переворачиваются на другую сторону).

В качестве фишек, меняющих свою форму на противоположную могут быть использованы так же и бокалы, которые могут находиться в двух положениях (ножками вверх и ножками вниз).

Бокалы – игра, задача которой заключается в том, чтобы перевернуть все бокалы ножками вверх за минимальное количество ходов. Начальное положение бокалов – ножками вниз.

Сложность игры в том, что одновременное можно перевернуть только определенное количество бокалов, которое не кратно количеству бокалов.

Актуальность игры заключается в том, чтобы человек развивал свое мышление и умение выбора правильного хода, суметь разработать алгоритм, по которому он будет играть и выигрывать.

Цель: Создание игры “Бокалы” в IDE “Lazarus”.

Задача: Научиться создавать игры в различных средах разработки.

# **Обзор аналогов**

**Реверси** (другое название — **оте́лло**) — настольная игра для двух человек на доске 8 на 8 клеток.

В игре используется квадратная доска размером 8 × 8 клеток (все клетки могут быть одного цвета) и 64 специальные фишки, окрашенные с разных сторон в контрастные цвета, например, в белый и чёрный. Клетки доски нумеруются от верхнего левого угла: вертикали — латинскими буквами, горизонтали — цифрами (по сути дела, можно использовать шахматную доску). Один из игроков играет белыми, другой — чёрными. Делая ход, игрок ставит фишку на клетку доски «своим» цветом вверх.

В начале игры в центр доски выставляются 4 фишки: чёрные на *d5* и *e4,* белые на *d4* и *e5.*

* Первый ход делают **чёрные**. Далее игроки ходят по очереди.
* Делая ход, игрок должен поставить свою фишку на одну из клеток доски таким образом, чтобы между этой поставленной фишкой и одной из имеющихся уже на доске фишек его цвета находился непрерывный ряд фишек соперника, горизонтальный, вертикальный или диагональный (другими словами, чтобы непрерывный ряд фишек соперника оказался «закрыт» фишками игрока с двух сторон). Все фишки соперника, входящие в «закрытый» на этом ходу ряд, переворачиваются на другую сторону (меняют цвет) и переходят к ходившему игроку.
* Если в результате одного хода «закрывается» одновременно более одного ряда фишек противника, то переворачиваются все фишки, оказавшиеся на всех «закрытых» рядах.
* Игрок вправе выбирать любой из возможных для него ходов. Если игрок имеет возможные ходы, он не может отказаться от хода. Если игрок не имеет допустимых ходов, то ход передаётся сопернику.
* Игра прекращается, когда на доску выставлены все фишки или когда ни один из игроков не может сделать хода. По окончании игры проводится подсчёт фишек каждого цвета, и игрок, чьих фишек на доске выставлено больше, объявляется победителем. В случае равенства количества фишек засчитывается ничья.

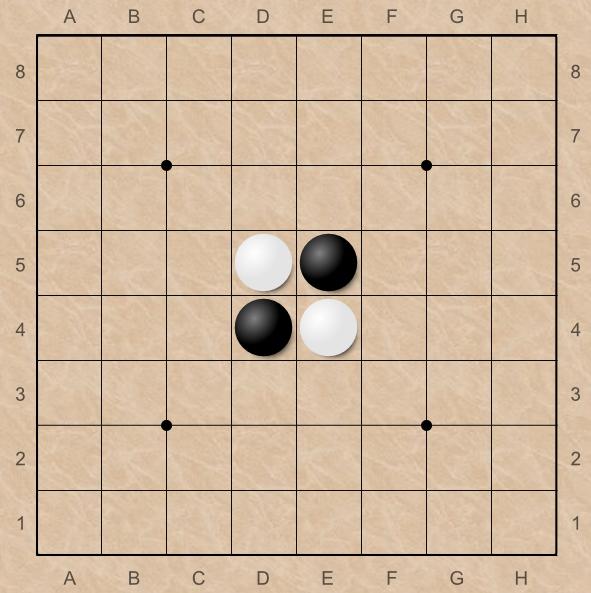


Рис.1 – Игровое поле игры “Реверси”.

**Flip-Flop -** Перед взором игрока предстает фасад здания, естественно, с окнами, в каждом из которых стоит шарик-помощник. Смело нажимайте на одного из них! В результате этого действия все соседи (а это и вертикальный, и горизонтальный ряды, на пересечении которых и находится выбранное окно) поменяют положение створок: открытые окна будут захлопнуты расстроенными хозяевами, а закрытые распахнутся навстречу свежему воздуху! Наша задача - проветрить весь дом, то есть привести все окна в открытое положение, ограничившись сотней действий. Каждый сотый клик возвращает игрока к началу.



Рис.2 – Интерфейс игры Flip-Flop.

# **Постановка задачи и требования, предъявляемые к программе**

Задачи:

1. Проанализировать аналоги игр-перевертышей
2. Перенести код программы из Delphi в Lazarus
3. Провести тестирование

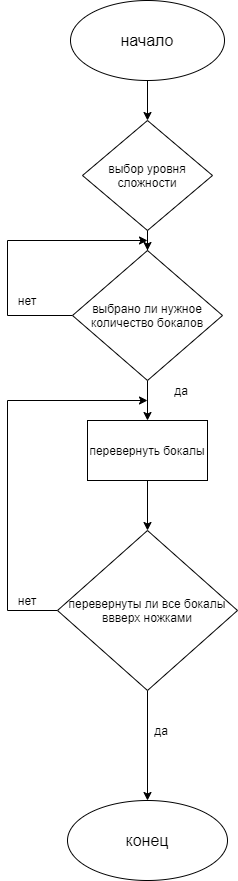
Требования:

1. Простота в освоении
2. Добавление дополнительных функций

# **Описание алгоритма работы программы**

1. При открытии приложения пользователю предлагается выбрать общее количество бокалова и количество которое нужно перевернуть.

2. Игра продолжается до того пока пользователь не перевернет все бокалы ножками вверх.

****

# **Выбор и обоснование выбора средства разработки, исходный код программы**

1. **Выбор средства разработки**

В качестве среды разработки была выбрана IDE Lazarus, так как очень удобна в написании кода, простой графический интерфейс, удобный вывод на форму, легко добавляемые элементы (кнопки, слайдеры и др.), в которые нужно просто написать код и привязать к нужному элементу.

Lazarus — открытая среда разработки программного обеспечения на языке Object Pascal для компилятора Free Pascal (часто используется сокращение FPC — Free Pascal Compiler, бесплатно распространяемый компилятор языка программирования Pascal). Интегрированная среда разработки предоставляет возможность кроссплатформенной разработки приложений в Delphi-подобном окружении.

1. **Исходный код**

unit Unit1;

interface

uses

LCLIntf, LCLType, LMessages, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

ExtCtrls, StdCtrls, Buttons;

const

NAME\_PROG = 'Ѕокалы';

type

TForm1 = class(TForm)

ImgUp: TImage;

ImgUpRed: TImage;

ImgDown: TImage;

imgDownRed: TImage;

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

Image7: TImage;

Image8: TImage;

Image9: TImage;

Image10: TImage;

Image11: TImage;

Image12: TImage;

Image13: TImage;

Image14: TImage;

Image15: TImage;

Image16: TImage;

Image17: TImage;

Image18: TImage;

Image19: TImage;

sbBok: TScrollBar;

sbPerev: TScrollBar;

Edit1: TEdit;

lblBok: TLabel;

lblPerev: TLabel;

Label2: TLabel;

Label4: TLabel;

Label5: TLabel;

sbtPerev: TSpeedButton;

sbtNewPlay: TSpeedButton;

Label1: TLabel;

lblHod: TLabel;

sbtHint: TSpeedButton;

procedure FormCreate(Sender: TObject);

procedure sbBokChange(Sender: TObject);

procedure sbPerevChange(Sender: TObject);

procedure Image1Click(Sender: TObject);

procedure sbtPerevClick(Sender: TObject);

procedure sbtNewPlayClick(Sender: TObject);

// procedure SpeedButton1Click(Sender: TObject);

procedure sbtHintClick(Sender: TObject);

// procedure ScrollBar1Change(Sender: TObject);

// procedure ListBox1Click(Sender: TObject);

private

{ Private declarations }

procedure ChangeBok;

procedure DrawBok;

procedure NewPlay;

// procedure Hint2;

procedure Hint;

public

{ Public declarations }

end;

type stBok= (Up, Down, UpRed, DownRed, None);

var

Form1: TForm1;

//всего бокалов:

nBok: integer;

//переворачиваем за один раз:

nPerev: integer;

//массив бокалов:

masBok: array[1..19] of stBok;

//ходы:

Hod: integer=0;

implementation

{$R \*.lfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

form1.AutoSize:= True;

NewPlay;

end; //FormCreate

procedure TForm1.DrawBok;

var

i: integer;

comp: TComponent;

begin

for i:= 1 to 19 do begin

comp:= FindComponent('Image'+inttostr(i));

(comp as Timage).top:= 36;

case masBok[i] of

None: (comp as Timage).visible:= False;

Up: begin

(comp as Timage).visible:= True;

(comp as Timage).Picture.Assign(ImgUp.Picture);

end;

Down: begin

(comp as Timage).visible:= True;

(comp as Timage).Picture.Assign(ImgDown.Picture);

end;

UpRed: begin

(comp as Timage).visible:= True;

(comp as Timage).Picture.Assign(ImgUpRed.Picture);

end;

DownRed: begin

(comp as Timage).visible:= True;

(comp as Timage).Picture.Assign(ImgDownRed.Picture);

end;

end;

end;

end; //DrawBok

procedure TForm1.ChangeBok;

var

i: integer;

begin

for i:= 1 to 19 do begin

if i <= nBok then

masBok[i]:= Up

else

masBok[i]:= None;

end;

DrawBok;

end; //ChangeBok

procedure TForm1.sbBokChange(Sender: TObject);

begin

lblBok.Caption:= inttostr(sbBok.Position);

nBok:= sbBok.Position;

//скорректировать число переворачиваемых бокалов:

sbPerevChange(Self);

end; //sbBokChange

procedure TForm1.sbPerevChange(Sender: TObject);

begin

lblPerev.Caption:= inttostr(sbPerev.Position);

nPerev:= sbPerev.Position;

while nPerev >= nBok do sbPerev.Position:= sbPerev.Position-1;

NewPlay;

end; //sbPerevChange

procedure TForm1.Image1Click(Sender: TObject);

var i: integer;

begin

//номер бокала:

i:= (Sender as TImage).tag;

case masBok[i] of

Up: masBok[i] := UpRed;

Down: masBok[i] := DownRed;

UpRed: masBok[i] := Up;

DownRed: masBok[i] := Down

end;

DrawBok;

end; //Image1Click

procedure TForm1.sbtPerevClick(Sender: TObject);

var i, n: integer;

begin

//проверить ход:

n:= 0;

for i:= 1 to 19 do

if (masBok[i]= UpRed) or (masBok[i]= DownRed)

then inc(n);

if n <> nPerev then begin

application.MessageBox('столько бокалов вам не перевернуть!',

NAME\_PROG,IDOK);

exit

end;

//выполнить ход:

for i:= 1 to 19 do

case masBok[i] of

UpRed: masBok[i] := Down;

DownRed: masBok[i] := Up

end;

DrawBok;

inc(Hod);

lblHod.Caption:= inttostr(Hod);

//все бокалы перевЄрнуты?

n:= 0;

for i:= 1 to 19 do

if masBok[i]= Down

then inc(n);

if n = nBok then begin

application.MessageBox('ќтлична¤ работа!',

NAME\_PROG,IDOK);

NewPlay;

end;

end; //sbtPerevClick

procedure TForm1.NewPlay;

begin

//число бокалов:

lblBok.Caption:= inttostr(sbBok.Position);

nBok:= sbBok.Position;

ChangeBok;

//переворачиваем за 1 раз:

lblPerev.Caption:= inttostr(sbPerev.Position);

nPerev:= sbPerev.Position;

//обнулить ходы:

Hod:= 0;

lblHod.Caption:= '0';

end; //NewPlay

procedure TForm1.sbtNewPlayClick(Sender: TObject);

begin

NewPlay;

end; //sbtNewPlayClick

procedure TForm1.SpeedButton1Click(Sender: TObject);

label

nextHod, nextPtr, BackHod;

var

str: array [0..20] of string;

i, n: integer;

maxHod: integer;

ptr: array[0..20] of integer;

Hod: integer;

s, strEnd: string;

begin

//ограничить глубину поиска:

maxHod:= 8;

maxHod:= scrollbar1.Position;

listbox1.Items.Add(inttostr(maxHod));

//listbox1.Items.Add(inttostr(nPerev));

//формируем начальную позицию:

s:= '';

for i:= 1 to 19 do

s:= s+ '1';

for i:= 0 to 20 do

str[i]:= s;

//формируем конечную позицию:

for i:= 1 to nBok do

s[i]:= '0';

strEnd:= s;

Hod:= 0;

//ptr[0]:= -1;

//сделать следующий ход:

nextHod:

inc(Hod);

if hod> maxHod then goto BackHod;

ptr[hod]:= 0;

//ptr[hod]:= ptr[hod-1]+1;

//переходим к следующей клетке:

nextPtr:

inc(ptr[hod]);

//if ptr[hod] > nBok - nPerev + 1 then goto backhod; //- дальше двигатьс¤ нельз¤

if ptr[hod] > nBok then goto backhod;

}

{//сделать ход - перевернуть бокалы:

str[hod]:= str[hod-1];

for i:= ptr[hod] to ptr[hod] + nPerev - 1 do

if str[hod-1][i]= '0' then

str[hod][i]:= '1'

else

str[hod][i]:= '0';}

{ //сделать ход - перевернуть бокалы:

str[hod]:= str[hod-1];

for i:= ptr[hod] to ptr[hod] + nPerev - 1 do

begin

if i> nBok then n:= i- nBok

else n:= i;

if str[hod-1][n]= '0' then

str[hod][n]:= '1'

else

str[hod][n]:= '0';

end;

s:= 'hod= '+inttostr(hod)+ ' '+ inttostr(ptr[hod])

+ ' позици¤: '+ str[hod];

//showmessage(s);

//listbox1.Items.Add(s);

//не решена ли задача?

if str[hod]= strEnd then //- решена!

begin

//записать решение в список:

listbox1.Items.Add('');

listbox1.Items.Add('');

for i:= 1 to hod do begin

s:= 'hod= '+inttostr(i)+ ' '+ inttostr(ptr[i])

+ ' позици¤: '+ str[i];

listbox1.Items.Add(s);

end;

showmessage('«адача решена!');

//exit;

//не искать длинных вариантов:

if hod< maxHod then maxHod:= hod;

goto backhod;

end;

//не повторилась ли позиции?

for i:= 0 to hod-1 do

if str[hod] = str[i] then //- была

goto nextptr;

goto nextHod;

//ход назад

BackHod:

dec(hod);

if hod<= 0 then begin

showmessage('¬сЄ!');

exit

end;

goto nextPtr;

end; }

{procedure TForm1.Hint2;

var

i, p, n: integer;

flg: Boolean;

begin

if odd(nBok) and not odd(nPerev) then

begin

showmessage('«адача решений не имеет!');

exit

end;

for i:= 0 to 100 do begin

if odd(nPerev) then

if odd(nBok) and not odd(i) then

else if not odd(nBok) and odd(i) then continue

else;

p:= (2+i)\*nPerev - nBok;

if (p mod 2 = 0) and (p>=0) then begin

p:= p div 2;

n:= i+2;

break;

end;

end;

showmessage('’одов: '+ inttostr (n)+ #10#13+'p = '+ inttostr (p));

end;}

procedure TForm1.Hint;

var

i, n, m: integer;

begin

//если число бокалов нечЄтное, а переворачиваем чЄтное

//число бокалов, то задача неразрешима:

if odd(nBok) and not odd(nPerev) then

begin

showmessage('«адача решений не имеет!');

exit

end;

m:= nBok;

n:= nPerev;

i:= 0;

while (n\*i< m) or ((n\*i - m) mod 2 <> 0) do inc (i);

showmessage(' решение задачи потребуется ходов - '+ inttostr (i)+

#10#13+

'число возвращаемых бокалов = '+ inttostr ((n \* i - m) div 2));

end; //Hint

//ѕќƒ— ј« ј

procedure TForm1.sbtHintClick(Sender: TObject);

begin

Hint;

end;

{procedure TForm1.ScrollBar1Change(Sender: TObject);

begin

label2.Caption:= inttostr(ScrollBar1.Position);

end;}

{procedure TForm1.ListBox1Click(Sender: TObject);

begin

ListBox1.Clear

end;}

end.

# **Тестирование программы**

**Чек-лист**

1. Осуществить поиск ошибок, в случае их наличия – исправить
2. Осуществлять неверные действия во время работы программы, чтобы отсмотреть ее поведение во время некорректных действий
3. Осуществить столько ходов, сколько бы человек не сделал и отследить наличие ошибок
4. Проверка на кроссплатформенность приложения

**Тестирование**

1. Было произведено тестирование кликером. Количество ходов превысило 1000 и приложение продолжало работать в штатном режиме.



Рис.3 – Результат тестирования кликером.

1. При неверном выборе количества переворачиваемых бокалов выходит сообщение: “Столько бокалов вам не перевернуть!”.

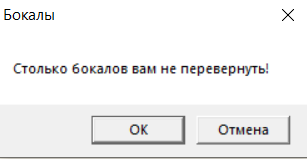


Рис.4 – Результат неверного выбора количества бокалов.

1. При прохождении игры выходит сообщение: “Отличная работа!”.

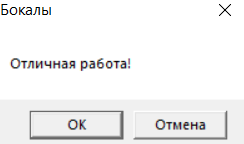


Рис.5 – Вывод сообщения при прохождении игры.

# **Руководство по установке**

Перейти по ссылке https://github.com/wagonaft/Bokali и скачать папку lib.

Разархивировать папку Lib в удобное для вас место и запустить exe файл Bokaly.

**Руководство пользователя**

Из папки “x86\_64-win64” запустить файл Bokaly.exe. Запустить программу кнопкой на панели инструментов “Запустить”, или клавишей F9.

После вывода формы предлагается выбрать то количество бокалов, которое нужно будет перевернуть и количество переворачиваемых бокалов одновременно. Далее можно начинать игру, выбираете бокалы, которые хотите перевернуть и нажимаете на кнопку “Перевернуть”. Главное – не бездумно переворачивать бокалы, продумайте алгоритм игры и решайте разного уровня задачи. Удачной игры!

**Заключение**

Во время прохождения учебной практики была выбрана тема “Создание игры головоломки-перевертыша Бокалы”. Были рассмотрены аналоги игры.

Произведен перевод исходного кода из IDE Delphi в IDE Lazarus.

Произведены различные виды тестирования, начиная от неправильных действий в игре (выбор неверного количества бокалов), продолжая тестированием программы на устойчивость к многочисленному количеству ходов, заканчивая прохождением игры.

Составлена блок-схема алгоритма работы программы.

Все цели и задачи практики, поставленные на время прохождения учебной практики были успешно выполнены и достигнуты.

**Список литературы**

1. http://www.freepascal.ru/download/pdf/osnovy\_programmirovanija\_v\_srede\_lazarus.pdf Мансуров К.Т. Основы программирования в среде Lazarus
2. http://www.mediafire.com/file/0yulzcjs4yd0wju/DelphiDop.rar/file# Delphi примерах, играх и программах. Дополнительные материал