Оглавление

[**1.** **Введение**](#_Toc407629101) 3

[**2.** **Основная часть**](#_Toc407629102) 4

[1. Спецификация](#_Toc407629103) 4

[2. Программа и методика испытаний](#_Toc407629104) 7

[3. Технический проект программного изделия](#_Toc407629105) 11

[4. Реализация программного изделия на языке программирования](#_Toc407629106) 14

[5. Тестирование программного продукта](#_Toc407629107) 15

[**3.** **Заключение**](#_Toc407629108) 16

[**4.** **Приложения**](#_Toc407629108) 18

[Приложение А](#_Toc407629107) 18

[Приложение Б](#_Toc407629107) 45

[Приложение В](#_Toc407629107) 46

# Введение

1. **Краткое описание сути темы курсового проекта**

Разработка игры «Тетрис». Игра должна создаваться с помощью программных средств среды Lazarus Forms, должно быть предусмотрено пять видов фигур. Более подробно описано в техническом задании.

1. **Обоснование актуальности и значимости темы**

Данная программа актуальна тем, что в процессе написания кода мною были изучены такие компоненты как TImage и TPanel. Так же повысился навык объектно-ориентированного программирования на язык Pascal и появилось желание писать большие проекты в будущем.

1. **Цели курсового проектирования в части формирования будущего специалиста**

Основной целью курсового проектирования является углубление и закрепление знаний по лекционному курсу, приобретение опыта самостоятельной работы по проектированию и написанию программного продукта и приобретение опыта работы со справочной литературой и ГОСТом.

# Основная часть

## **Спецификация**

Вначале разработки, были разработаны требования к программе.

1. Требования к программе
   1. Требования к функциональным характеристикам

Программа должна обеспечивать выполнение следующих основных функций:

* + 1. Функция «Интерактивная игра в «Тетрис» со стаканом (Высота 20, длина 10) ячеек»
    2. Функция «Движение фигур» При нажатии на кнопку стрелку влево фигура будет перемещаться на одну клетку влево, нажатие на кнопку стрелку вправо фигура будет перемещаться на одну клетку вправо, нажатии на кнопку стрелку вниз фигура будет перемещаться на одну клетку вниз до упора
    3. Функция «Смена координации фигуры» при нажатии на кнопку пробел, фигура будет поворачиваться на 90 градусов по часовой стрелке
    4. Функция «Таймер» отображает таймер времени, которое прошло сначала игры.
    5. Функция «Счетчик» отображает количество очков
    6. Функция «Результат игры» в конце игры выводится окно, в котором отображается затраченное время, и информация о количестве набранных очков и окно с просьбой ввести свое имя.
    7. Функция «Статистика» функция будет выводить результаты всех игр, которые были сыграны ранее.
    8. Функция «Сохранение результатов» функция будет сохранять результаты всех ранее сыгранных игр.
    9. Функция «Удаление вертикальных строк» удаляет вертикальную строку от левой границы до правой и увеличивает количество очков на единицу. И перемещает все фигуры сверху на одну ячейку вниз.
    10. Функция «Оповещения о следующей фигуре» рядом со стаканом будет выводиться поле с картинкой следующей фигуры. Пользователю будет разрешено отключить оповещение в настройке.

1. Требования к надежности
   1. Требования к контролю входной информации.
   2. Контролировать, наличие ошибок при работе с файлами.
   3. Контролировать, не ввел ли пользователь пустое имя.
   4. Контролировать, не выходит ли фигура за границы «Стакана»
   5. Контролировать, не наезжают ли фигуры друг на друга
   6. Условия эксплуатации(включая квалификацию персонала)
2. Условия эксплуатации
   1. Максимальное количество персонала – один человек, владеющий знаниями игры «Тетрис»
   2. Другие специальные требования к условиям эксплуатации не предъявляются
3. Требования к составу и параметрам технических средств

В состав технических средств должен входить IBM-совместимый компьютер, включающий:

* процессор Pentium или совместимый с ним;
* клавиатуру;
* видеокарту;
* монитор;
* жесткий или SSD-диск.

1. Требования к информационной и программной совместимости(требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой)
   1. Программа должна работать под управлением операционной системы Windows XP/7.
   2. Требования к защите информации и программ не предъявляются
2. Требования к интерфейсу:

* Общепринятый интерфейс для ОС windows XP/7
* Обычное меню
* Наличие ToolBar

## **Программа и методика испытаний**

Проверка осуществляется путем запуска программы и сравнения результатов ее запуска с ожидаемыми. Проверка игр должна проводится более 5 раз, но менее 10 раз.

При каждом запуске программы визуально контролируются совпадения результатов с ожидаемыми.

2.1 Метод проверки требований к программе

* В случае, если при начале игры запускается таймер времени и останавливается при поражении, то в протокол вносится запись: «Требования к функции «Таймер» - соответствует; в противном случае: «Требования к функции «Таймер»» - не соответствует.
* В случае, если при завершении игры выводится окно "Результат игры" с информацией о проигрыше, которое содержит информацию о счете, которое соответствует функции Счетчик, времени игры, которое соответствует функции таймер и имени игрока, которое вводится в конце, то в протокол вносится запись: «Требования к функции «Результат игры»» - соответствует; в противном случае: «Требования к функции «Результат игры»» - не соответствует.
* В случае, если результаты раннее сыгранных игр, сохраненных в специальный файл совпадают с результатами, выведенными на экран, в протокол вносится запись: «Требования к функции «Сохранение результатов» и «Статистика»» - соответствует; в противном случае: «Требования к функции «Сохранение результатов» и «Статистика»» - не соответствует.
* В случае, если рядом со стаканом выводится хотя бы один раз картина не со следующей фигурой, то в протокол носится запись: «Требования к функции «Оповещения о следующей фигуре» -не соответствует; в противном случае: «Требования к функции «Оповещения о следующей фигуре»» - соответствует
* В случае, если хот бы один раз при нажатии клавиши пробел фигура поворачивается на 90 градусов, и фигура находится на расстоянии своей длины от правого края стакана, то в протокол вносится запись: «Требования к функции «Смена координации»» -соответствует; в противном случае: «Требования к функции «Смена координации» -не соответствует».
* В случае, если хотя бы один раз удаление вертикальной строки происходит без увеличение счётчика на единицу, то в протокол вносится запись: «Требования к функции «Счетчик»» - не соответствует; в противном случае в протокол заносится запись «Требования к функции «Счетчик»» -соответствует.
* В случае, если хотя бы один раз вертикальная строка от левой до правой границе стакана не удаляется, то в протокол вносится запись: «Требования к функции «Удаление вертикальных строк» -не соответствует»; в противном случае: «Требования к функции «Удаление вертикальных строк» -соответствует».
* В случае, если программа не запускается на Windows Xp и на Windows 7, то в протокол вносится «Требования к информационной и программной совместимости» - не соответствует; в противном случае «Требования к информационной и программной совместимости» - соответствует.
* В случае, если

1. изменить права доступа к файлу с результатами на только для чтения и попробовать записать туда данные,
2. изменить права доступа к папке, в которой содержится файл с результатами и во время записи буду ошибки,
3. изменить права доступа к папке, в которой содержится файл с результатами и во время чтения будут ошибки,
4. вручную переименовать файл и при запуске программы,
5. открыть файл и изменить данные на некорректные и при чтение из фала

и на экран выводится сообщение об этой ошибке, то в протокол вносится «Требования к контролю входной информации «Контролировать наличие ошибок при работе с файлами»» - соответствуют; в ином случае «Требования к контролю входной информации «Контролировать наличие ошибок при работе с файлами»» - не соответствуют;

* В случае, если тестировщик оставил поле с именем пустым или заполнил имя одними пробелами и появилось сообщение об ошибке, то в протокол вносится: «Требования к контролю входной информации «Контролировать не ввел ли пользователь пустое имя»» - соответствуют; в ином случае то в протокол вносится: «Требования к контролю входной информации «Контролировать не ввел ли пользователь пустое имя»» - не соответствуют.
* В случае, если фигура выходит за границу «Стакана», то в протокол вносится «Требования к контролю входной информации «Контролировать не выходит ли фигура за границы «Стакана»»» - не соответствуют; в ином случае в протокол вносится «Требования к контролю входной информации «Контролировать не выходит ли фигура за границы «Стакана»»» - соответствуют.
* В случае, фигуры наезжают друг на друга, то в протокол вносится «Требования к контролю входной информации «Контролировать не наезжают ли фигуры друг на друга»» - не соответствуют; в ином случае в протокол вносится «Требования к контролю входной информации «Контролировать не наезжают ли фигуры друг на друга»» -соответствуют.
* Если на форме присутствует Toolbar и Обычное меню, то в протокол вносится «Требования к интерфейсу «Наличие Toolbar и Обычного меню»» -соответствует; иначе «Требования к интерфейсу «Наличие Toolbar и Обычного меню»» -не соответствует.
* Чтобы проверить информационную и программную совместимость нужно запустить программу сначала на Windows XP, а потом на Windows 7, то в протокол вносится запись: «Требования функции «Совместимость»» -соответствует; в противном случае: «Требования функции «Совместимость»» -не соответствует.
* Программа должна иметь общепринятый интерфейс Windows XP/7 ОС. Должно быть наличие обычного меню и ToolBar, причем компоненты обычного меню и меню ToolBar должны работать. Если каждое из этих условий реализуется, то в протокол вносится запись: «Требования к информационной и программной совместимости» - соответствует; в противном случае: «Требования к информационной и программной совместимости» - не соответствует. Пример кнопок интерфейса Приложение В.

## **Технический проект программного изделия**

1. **Обоснование выбора языка программирования**

Lazarus Free Pascal - выбран по причине простоты и скорости программирования, закрепления полученные знания, а также с целью познать тонкости работы с «событиями» и желания изучения компонентов и методов работы с ними, а также новых, для меня, встроенных стандартных модулей.

1. **Хранение данных**

Данные о «внутренностях стакана» хранятся в массиве math\_m(items).

type items=0..5;//Тип массива. Цифры означают цвета. 0-белый, 1-синий,2-темно-желтый, 3-зеленый, 4-фиолетовый, 5-светло-желтый.

Данные о движущейся фигуре хранятся в массиве figura(figurs)

type figures=record

x,y:integer;//Позиция для переноса массива с фигурой на массив с «внутренностях стакана»

vid:1..5;//Номер фигуры

color:1..5;//Номер цвета

math\_f:array[1..5,1..5] of integer;//Хранит фигуру.

**Блок-схема**

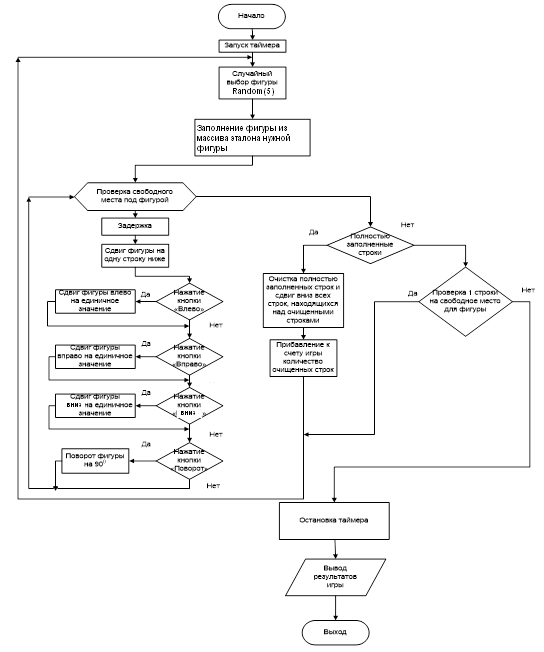


Рис.1

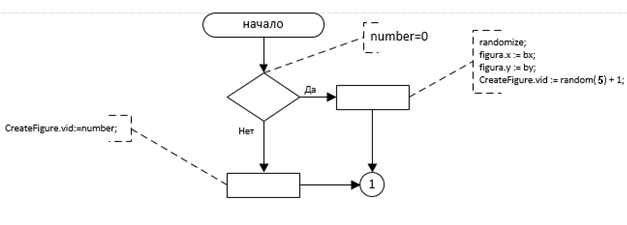


Рис.2

## **Реализация программного изделия на языке программирования**

Была разработана игра «Тетрис», исходный код в приложении А.

При разработки программного изделия сталкивался с множеством ошибок и трудностями. Ниже приведены некоторые ошибки и трудности, и то как преодолел их.

Некоторые из ошибок, с которыми сталкивался:

* 1. Не было добавлено в var переменной (добавил переменную в var)
  2. Синтаксические ошибки (исправлял синтаксис)

Некоторые трудности, с которыми сталкивался:

1. Создание фигуры вызывало критические ошибки (добавил ограничение для некоторых значений)
2. Удаление строки не переносило значения выше (изменил адресацию массива)
3. **Тестирование программного продукт**

Ниже приведена таблица тестирования программного продукта (таблица №1)

|  |  |
| --- | --- |
| Требования к функции «Таймер» | Соответствует |
| Требования к функции «Результат игры» | Соответствует |
| Требования к функции «Сохранение результатов» и «Статистика» | Соответствует |
| Требования к функции «Оповещения о следующей фигуре»» | Соответствует |
| Требования к функции «Смена координации» | Соответствует |
| Требования к функции «Счетчик» | Соответствует |
| Требования к функции «Удаление горизонтальных строк» | Соответствует |
| Требования к информационной и программной совместимости | Соответствует |
| Требования к контролю входной информации «Контролировать наличие ошибок при работе с файлами»» | соответствует |
| Требования к контролю входной информации «Контролировать не ввел ли пользователь пустое имя» | Соответствует |
| Требования к контролю входной информации «Контролировать не выходит ли фигура за границы «Стакана»» | Соответствует |
| Требования к контролю входной информации «Контролировать не наезжают ли фигуры друг на друга» | Не соответствует |
| Требования к интерфейсу «Наличие Toolbar и Обычного меню» | Соответствует |

## Таблица №1

## **6. Заключение**

1. **Насколько полно выполнен курсовой проект****.**

В ходе работы была создана игра «Тетрис» и комплекс стандартной документации к ней. Программа выполнена не полностью, имеются некоторые недочеты.

1. **Анализ эффективности и качества принятых решений во время курсового проектирования.**

Решения принятые в ходе работы были обдуманы и оговорены с руководителем проекта, в большинстве случаях верны.

1. **Выводы, полученные по результатам курсового проектирования****.**

В процессе выполнения курсовой работы получил опыт в долговременной разработке средних приложений. Освоил несколько новых компонентов в Lazarus Form, научился правильно распределять время разработки, закрепил знания, полученные ранее, и заниматься проектирование программы.

1. **Возможных планах по дальнейшей работе, связанной с темой курсового проектирования****.**

В дальнейшем в программу можно добавить новые фигуры, изменить дизайн.

**Список литературы**

[1] Topic: Memo. URL.

[http://](http://forum.lazarus.freepascal.org/index.php?topic=18853.0)www.delphi-manual.ru/memo.php

[2] Topic: TImage, 2014. URL.

<http://gospodaretsva.com/urok-3-komponent-timage.html>

[3] Topic: TPanel. URL.

[http://www.cyberguru.ru/delphi/delphi-components-part2- page35.html](http://www.cyberguru.ru/delphi/delphi-components-part2-page35.html)

[4] Lazarus – графический редактор на Lazarus’е, 2000-2014. URL.

<http://www.cyberforum.ru/lazarus/thread566537.html>

1. **Приложения**

*Приложение А*

Код программного продукта

*unit Unit1;*

*{$mode objfpc}{$H+}*

*interface*

*uses*

*crt, Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs,*

*ExtCtrls, LMessages, StdCtrls, Menus, ComCtrls, WinCrt;*

*type*

*{ Tetris }*

*StatGame = (nogame, partone, parttwo, partthree, partfour);*

*items = 0..5;*

*ttmas = array[1..5, 1..5] of integer;*

*figurs = record*

*x: integer;*

*y: integer;*

*vid: 1..5;*

*color: 1..5;*

*math\_f: ttmas;*

*end;*

*{ TTetris }*

*TTetris = class(TForm)*

*b\_OK: TButton;*

*History\_: TMemo;*

*ImageList1: TImageList;*

*ImgFon: TImage;*

*ImgBegin: TImage;*

*ImgSettings: TImage;*

*ImgResults: TImage;*

*ImgEND: TImage;*

*MainMenu1: TMainMenu;*

*MenuItem1: TMenuItem;*

*MenuItem2: TMenuItem;*

*MenuItem3: TMenuItem;*

*MenuItem4: TMenuItem;*

*MenuItem5: TMenuItem;*

*Name\_pol: TEdit;*

*GameTime: TPanel;*

*NextFigure: TPanel;*

*p\_text: TPanel;*

*p\_status: TPanel;*

*p\_GameOver: TPanel;*

*Score: TPanel;*

*Timer: TTimer;*

*Clock: TTimer;*

*ToolBar1: TToolBar;*

*ToolButton1: TToolButton;*

*ToolButton2: TToolButton;*

*ToolButton3: TToolButton;*

*ToolButton4: TToolButton;*

*procedure GameTimeClick(Sender: TObject);*

*procedure MenuItem2Click(Sender: TObject);*

*procedure MenuItem3Click(Sender: TObject);*

*procedure MenuItem4Click(Sender: TObject);*

*procedure MenuItem5Click(Sender: TObject);*

*procedure p\_GameOverClick(Sender: TObject);*

*procedure p\_statusClick(Sender: TObject);*

*procedure imgResultsClick(Sender: TObject);*

*procedure NextFigureClick(Sender: TObject);*

*procedure ClockTimer(Sender: TObject);*

*procedure p\_textClick(Sender: TObject);*

*procedure ScoreClick(Sender: TObject);*

*procedure TimerTimer(Sender: TObject);*

*procedure FormCreate(Sender: TObject);*

*procedure FormKeyDown(Sender: TObject; var Key: word; Shift: TShiftState);*

*procedure imgBeginClick(Sender: TObject);*

*procedure imgEndClick(Sender: TObject);*

*procedure imgSettingsClick(Sender: TObject);*

*procedure b\_OKClick(Sender: TObject);*

*procedure Move(Sender: TObject);*

*function bMove(x, y: integer): boolean;*

*function bSwap: boolean;*

*procedure GameEnd(Sender: TObject);*

*procedure delete\_line(Sender: TObject);*

*procedure swap(var mm: ttmas);*

*function CreateFigure(Sender: TObject): figurs;*

*procedure graphic\_(Sender: TObject);*

*procedure CloseGame;*

*procedure ToolButton1Click(Sender: TObject);*

*procedure ToolButton2Click(Sender: TObject);*

*procedure ToolButton3Click(Sender: TObject);*

*procedure ToolButton4Click(Sender: TObject);*

*private*

*{ private declarations }*

*public*

*{ public declarations }*

*MainFullFon: TBitmap;*

*math\_m: array[0..11, -4..21] of items;//Математическое представление*

*graphic\_m: array[1..10, 1..20] of TImage;//Графический вывод Готов*

*sled\_m: array[1..5, 1..5] of TImage;//Графический вывод Готов*

*figura: figurs;//Движущаяся фигура*

*sostgame: StatGame;//Этап игры*

*ClockSec, ClockHour: longint;//Время считать*

*sGameTime: string;*

*status: boolean;//В настройках изменяется*

*MovingFigura: boolean;//Фигура движется*

*number: integer;//Номер фигуры*

*points: longint;//Счет*

*sled: integer;*

*sledMas:ttmas;*

*sledcolor:tcolor;*

*bx, by: integer;*

*const*

*First: boolean = True;*

*end;*

*var*

*Tetris: TTetris;*

*etalon1: ttmas = ((0, 0, 1, 0, 0), (0, 0, 1, 0, 0), (0, 0, 1, 0, 0),*

*(0, 0, 1, 0, 0), (0, 0, 0, 0, 0));*

*etalon2: ttmas = ((0, 0, 0, 0, 0), (0, 0, 1, 0, 0), (0, 0, 1, 0, 0),*

*(0, 0, 1, 1, 0), (0, 0, 0, 0, 0));*

*etalon3: ttmas = ((0, 0, 0, 0, 0), (0, 0, 1, 1, 0), (0, 1, 1, 0, 0),*

*(0, 0, 0, 0, 0), (0, 0, 0, 0, 0));*

*etalon4: ttmas = ((0, 0, 0, 0, 0), (0, 0, 1, 0, 0), (0, 1, 1, 1, 0),*

*(0, 0, 0, 0, 0), (0, 0, 0, 0, 0));*

*etalon5: ttmas = ((0, 0, 0, 0, 0), (0, 1, 1, 0, 0), (0, 1, 1, 0, 0),*

*(0, 0, 0, 0, 0), (0, 0, 0, 0, 0));*

*implementation*

*{$R \*.lfm}*

*{ TTetris }*

*{ Tetris }*

*procedure TTetris.NextFigureClick(Sender: TObject);*

*begin*

*Timer.Enabled := False;*

*GameEnd(self);*

*delete\_line(self);*

*end;*

*{Действия после конца игры}*

*procedure TTetris.GameEnd(Sender: TObject);*

*var*

*i, j: integer;*

*begin*

*sostgame := nogame;*

*b\_OK.Visible := True;*

*p\_GameOver.Visible := True;*

*p\_GameOver.Caption := 'GAME OVER! Набрано очков ' + IntToStr(points) + ', время игры ' +*

*IntToStr(ClockHour) + ':' + IntToStr(ClockSec);*

*Name\_pol.Visible := True;*

*MovingFigura := False;*

*GameTime.Visible := False;*

*Score.Visible := False;*

*NextFigure.Visible := False;*

*timer.Enabled := False;*

*clock.Enabled := False;*

*p\_text.Visible := False;*

*p\_status.Visible := False;*

*for i := 1 to 10 do*

*for j := 1 to 20 do*

*begin*

*graphic\_m[i, j].Visible := False;*

*end;*

*for i := 1 to 10 do*

*for j := -4 to 20 do*

*begin*

*math\_m [i, j]:=0;*

*end;*

*Score.Caption := '';*

*sGameTime := '';*

*end;*

*{-------------------------}*

*{Действия после нажатия на ОК}*

*procedure TTetris.b\_OKClick(Sender: TObject);*

*var*

*s: string;*

*f: TextFile;*

*begin*

*{$I-}*

*AssignFile(f, 'Histrory.txt');*

*if FileExists('Histrory.txt') = True then*

*begin*

*if (Length(Name\_pol.Text) = 0) or (Name\_pol.Text = ' ') then*

*ShowMessage('Пустое имя запрещено! Результаты сохранены не будут! ')*

*else*

*begin*

*s := Name\_pol.Text;*

*s := '| ' + s + ' | ' + IntToStr(points) + ' | ' + IntToStr(ClockHour) +*

*':' + IntToStr(ClockSec) + ' |';*

*append(f);*

*writeln(f, s);*

*CloseFile(f);*

*end;*

*end*

*else*

*ShowMessage('Файл с результатами прошлых игр не найден');*

*{$I+}*

*GameTime.Caption := '';*

*imgBegin.Visible := True;*

*imgSettings.Visible := True;*

*imgResults.Visible := True;*

*imgEnd.Visible := True;*

*History\_.Visible := False;*

*b\_OK.Visible := False;*

*p\_GameOver.Visible := False;*

*Name\_pol.Visible := False;*

*end;*

*{-------------------------}*

*{Удаляет линию и сдвигает "остатки" от фигур}*

*procedure TTetris.delete\_line(Sender: TObject);*

*var*

*i, j, x, y: integer;*

*l: integer;*

*begin*

*begin*

*for j := 1 to 20 do*

*begin*

*l := 0;*

*for i := 1 to 10 do*

*begin*

*if math\_m[i, j] > 0 then*

*Inc(l);*

*end;*

*if l = 10 then*

*begin*

*if j <> 1 then*

*begin*

*for y := j downto 2 do*

*begin*

*for x := 1 to 10 do*

*begin*

*math\_m[x, y] := math\_m[x, y - 1];*

*end;*

*end;*

*end*

*else*

*for x := 1 to 10 do*

*math\_m[x, j] := 0;*

*Inc(points);*

*end;*

*end;*

*end;*

*end;*

*{-------------------------------------------}*

*{Действия при нажатии на кнопку РЕЗУЛЬТАТЫ}*

*procedure TTetris.imgResultsClick(Sender: TObject);*

*var*

*f: TextFile;*

*s: string;*

*begin*

*// if sostgame = nogame then*

*CloseGame;*

*begin*

*sostgame := partthree;*

*History\_.Visible := True;*

*imgBegin.Visible := False;*

*imgSettings.Visible := False;*

*imgResults.Visible := False;*

*imgEnd.Visible := False;*

*{$I-}*

*AssignFile(f, 'Histrory.txt');*

*if FileExists('Histrory.txt') = True then*

*begin*

*Reset(f);*

*History\_.Lines.Clear;*

*while not (EOF(f)) do*

*begin*

*ReadLn(f, s);*

*History\_.Lines.Add(s);*

*end;*

*CloseFile(f);*

*end*

*else*

*ShowMessage('Файл с результатами прошлых игр не найден');*

*{$I+}*

*if IOResult <> 0 then*

*ShowMessage('Ошибка доступа к файлу');*

*end;*

*end;*

*{-----------------------------------------}*

*procedure TTetris.Move(Sender: TObject);*

*var*

*i, j: integer;*

*begin*

*for i := 1 to 5 do*

*for j := 1 to 5 do*

*begin*

*if math\_m[i, j] <> 0 then*

*begin*

*math\_m[i, j] := 0;*

*end;*

*end;*

*end;*

*function TTetris.bMove(x, y: integer): boolean;*

*var*

*i, j: integer;*

*begin*

*bMove := True;*

*for i := 1 to 5 do*

*for j := 1 to 5 do*

*if figura.math\_f[i, j] = 1 then*

*if math\_m[figura.x + i + x, figura.y + j + y] <> 0 then*

*begin*

*bMove := False;*

*break;*

*end;*

*end;*

*function TTetris.bSwap: boolean;*

*var*

*i, j: integer;*

*tmp: ttmas;*

*begin*

*bSwap := True;*

*tmp := figura.math\_f;*

*swap(tmp);*

*for i := 1 to 5 do*

*for j := 1 to 5 do*

*if tmp[i, j] = 1 then*

*if math\_m[figura.x + i, figura.y + j] <> 0 then*

*begin*

*bSwap := False;*

*break;*

*end;*

*end;*

*{Таймер для обновления картинок}*

*procedure TTetris.TimerTimer(Sender: TObject);*

*var*

*i, j, l, k: integer;*

*s: string;*

*begin*

*if (sostgame = partone) then*

*begin*

*if bMove(0, 1) then*

*Inc(figura.y); //Down*

*if bMove(0, 1) = False then*

*begin*

*for i := 1 to 5 do*

*for j := 1 to 5 do*

*if figura.math\_f[i, j] = 1 then*

*math\_m[figura.x + i, figura.y + j] := figura.color;*

*for i:=1 to 10 do*

*if math\_m[i,1]<>0 then*

*begin*

*GameEnd(self);*

*break;*

*end;*

*begin*

*number := 0;*

*CreateFigure(self);*

*end;*

*end;*

*end;*

*delete\_line(self);*

*Score.Caption := IntToStr(points);*

*graphic\_(self);*

*end;*

*{------------------------------}*

*{Нажатие на кнопку СТАРТ}*

*procedure TTetris.imgBeginClick(Sender: TObject);*

*{Заполняет картинки белым}*

*procedure GoWhite(Sender: TObject);*

*var*

*x, y: integer;*

*begin*

*for x := 1 to 10 do*

*for y := 1 to 20 do*

*graphic\_m[x, y].Visible := True;*

*end;*

*begin*

*// if sostgame = nogame then*

*CloseGame;*

*begin*

*sostgame := partone;*

*imgBegin.Visible := False;*

*imgSettings.Visible := False;*

*imgResults.Visible := False;*

*imgEnd.Visible := False;*

*GameTime.Visible := True;*

*Score.Visible := True;*

*nextfigure.visible:= status;*

*History\_.Visible := False;*

*GoWhite(self);*

*number := 0;*

*CreateFigure(self);*

*MovingFigura := True;*

*Timer.Enabled := true;*

*Clock.Enabled := true;*

*end;*

*end;*

*{-----------------------}*

*{Нажатие на кнопку ВЫХОД}*

*procedure TTetris.imgEndClick(Sender: TObject);*

*begin*

*Close;*

*end;*

*{-----------------------}*

*{Нажатие на кнопку НАСТРОЙКИ}*

*procedure TTetris.imgSettingsClick(Sender: TObject);*

*begin*

*// if sostgame = nogame then*

*CloseGame;*

*begin*

*sostgame := parttwo;*

*imgBegin.Visible := False;*

*imgSettings.Visible := False;*

*imgResults.Visible := False;*

*imgEnd.Visible := False;*

*p\_text.Caption := 'Оповещать о следующей фигуре ';*

*p\_text.Visible := True;*

*p\_status.Visible := True;*

*if status = True then*

*p\_status.Caption := 'Да'*

*else*

*p\_status.Caption := 'Нет';*

*end;*

*end;*

*{---------------------------}*

*{Нажатие на кнопку НАСТРОЙКИ.ОПЕВЕЩЕНИЕ О СЛЕДУЮЩЕЙ ФИГУРЕ}*

*procedure TTetris.p\_statusClick(Sender: TObject);*

*begin*

*if status = True then*

*begin*

*p\_status.Caption := 'Нет';*

*status := False;*

*end*

*else*

*begin*

*p\_status.Caption := 'Да';*

*status := True;*

*end;*

*end;*

*procedure TTetris.p\_GameOverClick(Sender: TObject);*

*begin*

*end;*

*procedure TTetris.GameTimeClick(Sender: TObject);*

*begin*

*end;*

*procedure TTetris.MenuItem2Click(Sender: TObject);*

*begin*

*close;*

*end;*

*procedure TTetris.MenuItem3Click(Sender: TObject);*

*begin*

*imgResultsClick(self);*

*end;*

*procedure TTetris.MenuItem4Click(Sender: TObject);*

*begin*

*imgSettingsClick(self);*

*end;*

*procedure TTetris.MenuItem5Click(Sender: TObject);*

*begin*

*imgBeginClick(self);*

*end;*

*{---------------------------------------------------------}*

*{Создание формы}*

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

*var*

*x, y: integer;*

*begin*

*DoubleBuffered := True;*

*number := 0;*

*points := 0;*

*History\_.Visible := False;*

*for x := 1 to 5 do*

*begin*

*for y := 1 to 5 do*

*begin*

*sled\_m[x, y] := TImage.Create(nextfigure);*

*sled\_m[x, y].parent := nextfigure;*

*sled\_m[x, y].Visible := true;*

*sled\_m[x, y].Left := 2+30 \* (x-1);*

*sled\_m[x, y].Top := 2+30 \* (y-1);*

*sled\_m[x, y].Picture.LoadFromFile('./TetrisImages/whiteBlock.png');*

*end;*

*end;*

*for x := 1 to 10 do*

*begin*

*for y := 1 to 20 do*

*begin*

*math\_m[x, y] := 0;*

*graphic\_m[x, y] := TImage.Create(self);*

*graphic\_m[x, y].Parent := Self;*

*graphic\_m[x, y].Visible := False;*

*graphic\_m[x, y].Left := 30 \* x;*

*graphic\_m[x, y].Top := 30 \* y;*

*graphic\_m[x, y].Picture.LoadFromFile('./TetrisImages/whiteBlock.png');*

*end;*

*end;*

*for y := 1 to 20 do //рамки*

*begin*

*math\_m[0, y] := -1;*

*math\_m[11, y] := -1;*

*end;*

*for x := 1 to 10 do*

*math\_m[x, 21] := -1;*

*number := 0;*

*b\_OK.Visible := False;*

*p\_GameOver.Visible := False;*

*Name\_pol.Visible := False;*

*MovingFigura := False;*

*status := True;*

*ClockSec := 0;*

*ClockHour := 0;*

*sostgame := nogame;*

*imgFon.Picture.Bitmap.LoadFromFile('./TetrisImages/FullFon.bmp');*

*ImgBegin.Picture.Bitmap.LoadFromFile('./TetrisImages/1.bmp');*

*ImgSettings.Picture.Bitmap.LoadFromFile('./TetrisImages/2.bmp');*

*ImgResults.Picture.Bitmap.LoadFromFile('./TetrisImages/3.bmp');*

*ImgEnd.Picture.Bitmap.LoadFromFile('./TetrisImages/4.bmp');*

*end;*

*{--------------}*

*{Таймер для подсчета времени игры}*

*procedure TTetris.ClockTimer(Sender: TObject);*

*begin*

*if ClockSec < 59 then*

*Inc(ClockSec)*

*else*

*begin*

*Inc(ClockHour);*

*ClockSec := 0;*

*end;*

*sGameTime := 'Время игры ' + IntToStr(ClockHour) + ':' + IntToStr(ClockSec);*

*GameTime.Caption := sGameTime;*

*end;*

*procedure TTetris.p\_textClick(Sender: TObject);*

*begin*

*end;*

*procedure TTetris.ScoreClick(Sender: TObject);*

*begin*

*end;*

*{--------------------------------}*

*{Конвертирование из математического представления в графическое}*

*procedure TTetris.graphic\_(Sender: TObject);*

*var*

*i, j: integer;*

*begin*

*for i := 1 to 10 do*

*for j := 1 to 20 do*

*begin*

*graphic\_m[i, j].Picture.Clear;*

*case math\_m[i, j] of*

*0: graphic\_m[i, j].Picture.LoadFromFile('./TetrisImages/whiteBlock.png');*

*1: graphic\_m[i, j].Picture.LoadFromFile('./TetrisImages/greenBlock.png');*

*2: graphic\_m[i, j].Picture.LoadFromFile('./TetrisImages/redBlock.png');*

*3: graphic\_m[i, j].Picture.LoadFromFile('./TetrisImages/blueBlock.png');*

*4: graphic\_m[i, j].Picture.LoadFromFile('./TetrisImages/yeBlock.png');*

*5: graphic\_m[i, j].Picture.LoadFromFile('./TetrisImages/yellowBlock.png');*

*end;*

*end;*

*for i := 1 to 5 do*

*for j := 1 to 5 do*

*if figura.math\_f[i, j] = 1 then*

*if ((i + figura.x) in [1..10]) and ((j + figura.y) in [1..20]) then*

*case figura.color of*

*0: graphic\_m[i + figura.x, j + figura.y].Picture.LoadFromFile(*

*'./TetrisImages/whiteBlock.png');*

*1: graphic\_m[i + figura.x, j + figura.y].Picture.LoadFromFile(*

*'./TetrisImages/greenBlock.png');*

*2: graphic\_m[i + figura.x, j + figura.y].Picture.LoadFromFile(*

*'./TetrisImages/redBlock.png');*

*3: graphic\_m[i + figura.x, j + figura.y].Picture.LoadFromFile(*

*'./TetrisImages/blueBlock.png');*

*4: graphic\_m[i + figura.x, j + figura.y].Picture.LoadFromFile(*

*'./TetrisImages/yeBlock.png');*

*5: graphic\_m[i + figura.x, j + figura.y].Picture.LoadFromFile(*

*'./TetrisImages/yellowBlock.png');*

*end;*

*for i := 1 to 5 do*

*for j := 1 to 5 do*

*begin*

*sled\_m[i, j].Picture.Clear;*

*if sledmas[i, j]=0 then*

*sled\_m[i, j].Picture.LoadFromFile('./TetrisImages/whiteBlock.png')*

*else*

*case sled of*

*0: sled\_m[i, j].Picture.LoadFromFile('./TetrisImages/whiteBlock.png');*

*1: sled\_m[i, j].Picture.LoadFromFile('./TetrisImages/greenBlock.png');*

*2: sled\_m[i, j].Picture.LoadFromFile('./TetrisImages/redBlock.png');*

*3: sled\_m[i, j].Picture.LoadFromFile('./TetrisImages/blueBlock.png');*

*4: sled\_m[i, j].Picture.LoadFromFile('./TetrisImages/yeBlock.png');*

*5: sled\_m[i, j].Picture.LoadFromFile('./TetrisImages/yellowBlock.png');*

*end;*

*end;*

*end;*

*procedure TTetris.CloseGame;*

*var i,j:integer;*

*begin*

*MovingFigura := False;*

*GameTime.Visible := False;*

*Score.Visible := False;*

*nextfigure.visible :=false;*

*NextFigure.Visible := False;*

*timer.Enabled := False;*

*clock.Enabled := False;*

*p\_text.Visible := False;*

*p\_status.Visible := False;*

*for i := 1 to 10 do*

*begin*

*for j := 1 to 20 do*

*begin*

*graphic\_m[i, j].Visible := False;*

*math\_m[i, j]:=0;*

*end;*

*end;*

*imgBegin.Visible := True;*

*imgSettings.Visible := True;*

*imgResults.Visible := True;*

*imgEnd.Visible := True;*

*sostgame := nogame;*

*Score.Caption := '';*

*GameTime.Caption := '';*

*sGameTime := '';*

*ClockSec := 0;*

*ClockHour := 0;*

*points:=0;*

*History\_.Visible := False;*

*b\_OK.Visible := False;*

*p\_GameOver.Visible := False;*

*Name\_pol.Visible := False;*

*end;*

*procedure TTetris.ToolButton1Click(Sender: TObject);*

*begin*

*imgBeginClick(self);*

*end;*

*procedure TTetris.ToolButton2Click(Sender: TObject);*

*begin*

*imgSettingsClick(self);*

*end;*

*procedure TTetris.ToolButton3Click(Sender: TObject);*

*begin*

*imgResultsClick(self);*

*end;*

*procedure TTetris.ToolButton4Click(Sender: TObject);*

*begin*

*close;*

*end;*

*{--------------------------------------------------------------}*

*{Функция для поворота массива с фигурой}*

*procedure TTetris.swap(var mm: ttmas);*

*var*

*i, j, n: integer;*

*tmp: ttmas;*

*begin*

*n := 5;*

*tmp := mm;*

*for j := 1 to n do*

*for i := 1 to n do*

*mm[i, j] := tmp[j, n - i + 1];*

*end;*

*{--------------------------------------}*

*{Функция для создания случайной фигуры}*

*function TTetris.CreateFigure(Sender: TObject): figurs;*

*begin*

*if number = 0 then*

*begin*

*randomize;*

*if First = True then*

*begin*

*First := False;*

*sled := random(5) + 1;*

*end;*

*number := sled;*

*sled := random(5) + 1;*

*end;*

*case number of*

*1: figura.color := 1;*

*2: figura.color := 2;*

*3: figura.color := 3;*

*4: figura.color := 4;*

*5: figura.color := 5;*

*end;*

*case number of*

*1: figura.math\_f := etalon1;*

*2: figura.math\_f := etalon2;*

*3: figura.math\_f := etalon3;*

*4: figura.math\_f := etalon4;*

*5: figura.math\_f := etalon5;*

*end;*

*case sled of*

*1: sledcolor := 1;*

*2: sledcolor := 2;*

*3: sledcolor := 3;*

*4: sledcolor := 4;*

*5: sledcolor := 5;*

*end;*

*case sled of*

*1: sledMas := etalon1;*

*2: sledMas := etalon2;*

*3: sledMas := etalon3;*

*4: sledMas := etalon4;*

*5: sledMas := etalon5;*

*end;*

*figura.x := 3;*

*figura.y := -3;*

*end;*

*{-------------------------------------}*

*{Действия при нажатии кнопок}*

*procedure TTetris.FormKeyDown(Sender: TObject; var Key: word; Shift: TShiftState);*

*var*

*i, j: integer;*

*begin*

*if (sostgame = partone) and (MovingFigura = True) then*

*begin*

*if (key = 32) and bSwap then*

*swap(figura.math\_f); //Space*

*if (key = 37) and bMove(-1, 0) then*

*Dec(figura.x); //Left*

*if (key = 39) and bMove(1, 0) then*

*Inc(figura.x); //Right*

*if (key = 40) and bMove(0, 1) then*

*Inc(figura.y); //Down*

*end;*

*if (sostgame <> nogame) and (key = 27) then*

*begin*

*CLoseGame;*

*end;*

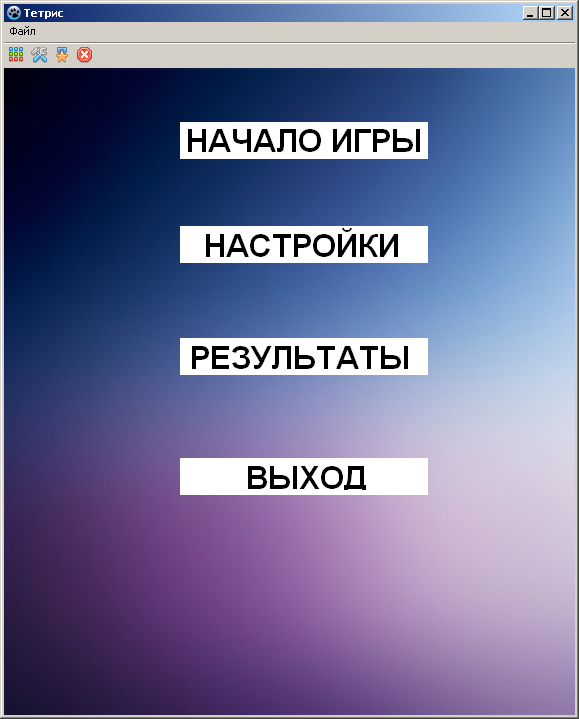
*end;*

*{---------------------------}*

*end.*

*Приложение Б*

Меню программы



*Приложение В*

Процесс игры

