

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Белгородский государственный технологический университет  
имени В.Г. Шухова

Институт ИЭИТУС  
Кафедра Информационных технологий

**Курсовой проект**  
**по дисциплине технологии программирования на тему:**

*«Разработка игровой программы «Колобок»*

Выполнил студент группы ИТ- 12

*Давыдов В.О.*

Проверил *к.т.н., доц..*

*Стативко Р.А.*

Белгород 2018

## Оглавление

Введение.....	3
1.Постановка задачи и основных требований к разрабатываемому программному обеспечению «Колобок».....	4
1.1. Постановка задачи.....	4
1.2. Основания для разработки.....	4
1.3. Назначение программного средства.....	4
1.4. Требования к программному средству «Колобок».....	4
1.4.1. Требования к функциональным характеристикам.....	4
1.4.2. Требования к надежности.....	5
1.4.3. Требования к составу и параметрам технических средств.....	5
1.5. Требования к условиям эксплуатации.....	5
1.6. Требования к программной документации.....	5
2. Проектирование программного средства и реализация программы «Колобок»	6
2.1. Разработка структурной схемы программы.....	6
2.2. Разработка алгоритмов программы.....	6
2.3. Описание структур, типов данных и глобальных переменных.....	23
2.4. Описание процедур и функций.....	24
Заключение.....	32
Список литературы.....	33
Приложение 1 Листинг программы.....	33
Приложение 2 Скриншоты программы.....	51

## **Введение**

В современном мире существует большое разнообразие компьютерных игр. Они получили большую популярность и нашли поклонников по всему миру. Игры отличаются по жанру, способам и методам написания, применением различных технологий, для различных устройств.

В настоящее время игры – один из самых популярных способов скоротать свой досуг. Год от года требования игроков растут, по этой причине игры становятся все более реалистичными и сложными, требующими все более мощных аппаратных ресурсов.

Настоящее техническое задание направлено на разработку простой игровой программы «Колобок» на языке высокого уровня Pascal.

# **1. Постановка задачи и основных требований к разрабатываемому программному обеспечению «Колобок»**

## **1.1. Постановка задачи**

Основание для проведения разработки является задание по курсовому проектированию, которое предполагает разработку программного средства, согласно своему варианту:

*По дороге движется колобок. По пути встречаются препятствия. Одни из них являются «положительными» и увеличивают число очков, другие же наоборот – являются «отрицательными» и уменьшают очки. Колобок может прыгать, ускорять и замедлять движение. Пользователю предлагается выбрать количество каждых препятствий. По завершении игры результаты сохраняются в файл.*

Данный программный продукт должен иметь интерфейс понятный пользователю практически не знакомого с персональной техникой, управление осуществляется с клавиатуры.

## **1.2. Основания для разработки**

Программное средство разрабатывается на основе учебного плана кафедры «Информационные технологии» для специальности 09.03.02 «Информационные системы и технологии» по предмету «Технологии программирования».

## **1.3. Назначение программного средства**

Программа представляет собой компьютерную игру «Колобок» и направлена на организацию досуга пользователя.

## **1.4. Требования к программному средству «Колобок»**

### **1.4.1. Требования к функциональным характеристикам**

#### **1.4.1.1. Программа должна обеспечивать возможность выполнения следующих функций**

- обеспечение эффективного диалога с пользователем;

- выбор количества «положительных» и «отрицательных» препятствий;
- проведение игрового процесса;
- сохранение результатов игры.

#### **1.4.1.2. Исходные данные**

- Начало новой игры;
- Выбор количества «положительных» препятствий;
- Выбор количества «отрицательных» препятствий;
- Просмотр инструкций;
- Просмотр топ игроков;
- Выход из программы;

#### **1.4.2. Требования к надежности**

Предусмотреть блокировку некорректных действий пользователя при работе с системой, предусмотреть контроль вводимой пользователем информации. Программа должна функционировать корректно, в соответствии с разработанным алгоритмом.

#### **1.4.3. Требования к составу и параметрам технических средств**

Необходимо наличие IBM PC - совместимого ПК с графическим адаптером EGA (VGA). Дисковое пространство должно быть не менее 100 Кб, объем свободной оперативной памяти - не менее 800 Кб. Желательно наличие драйвера EMS. Наличие манипулятора типа "клавиатура".

#### **1.5. Требования к условиям эксплуатации**

Для запуска файла программы, на компьютере должен быть установлен компилятор языка Pascal «Turbo Pascal». Для поддержки функции сохранения результатов в папке «Мои документы» должна существовать папка «kolobok» с текстовым файлом «result» внутри. Исправно функционирующие комплектующие компьютера.

#### **1.6. Требования к программной документации**

Необходимы спецификации программ, содержащие описания целей создания переменных и блок-схемы.

Разрабатываемые программные модули должны быть самодокументированы, т. е. тексты программ должны содержать все необходимые комментарии.

## 2. Проектирование программного средства и реализация программы «Колобок»

### 2.1. Разработка структурной схемы программы



Рисунок 1 Структурная схема программы

### 2.2. Разработка алгоритмов программы



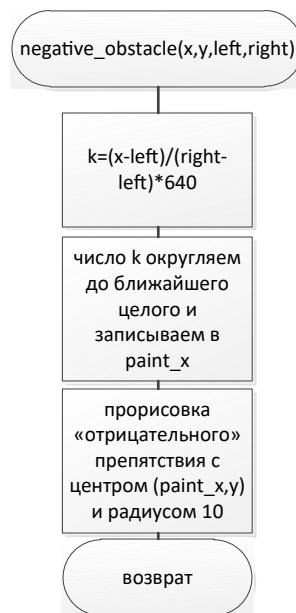
Рисунок 2 Процедура прорисовки приветственного экрана



**Рисунок 3 Процедура прорисовки подтверждения выхода из игры**



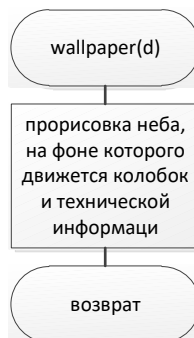
**Рисунок 4 Процедура, рисующая коlobка**



**Рисунок 5 Процедура, рисующая «отрицательное препятствие»**

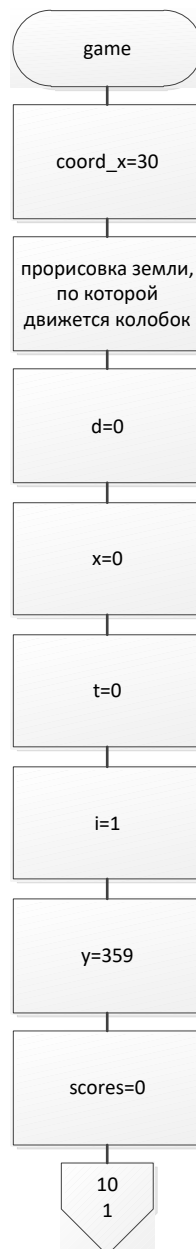


**Рисунок 6 Процедура, рисующая «положительное» препятствие**



**Рисунок 7 Процедура, рисующая небо, на фоне которого движется колобок, и техническую информацию**





**Рисунок 8 Процедура игры**

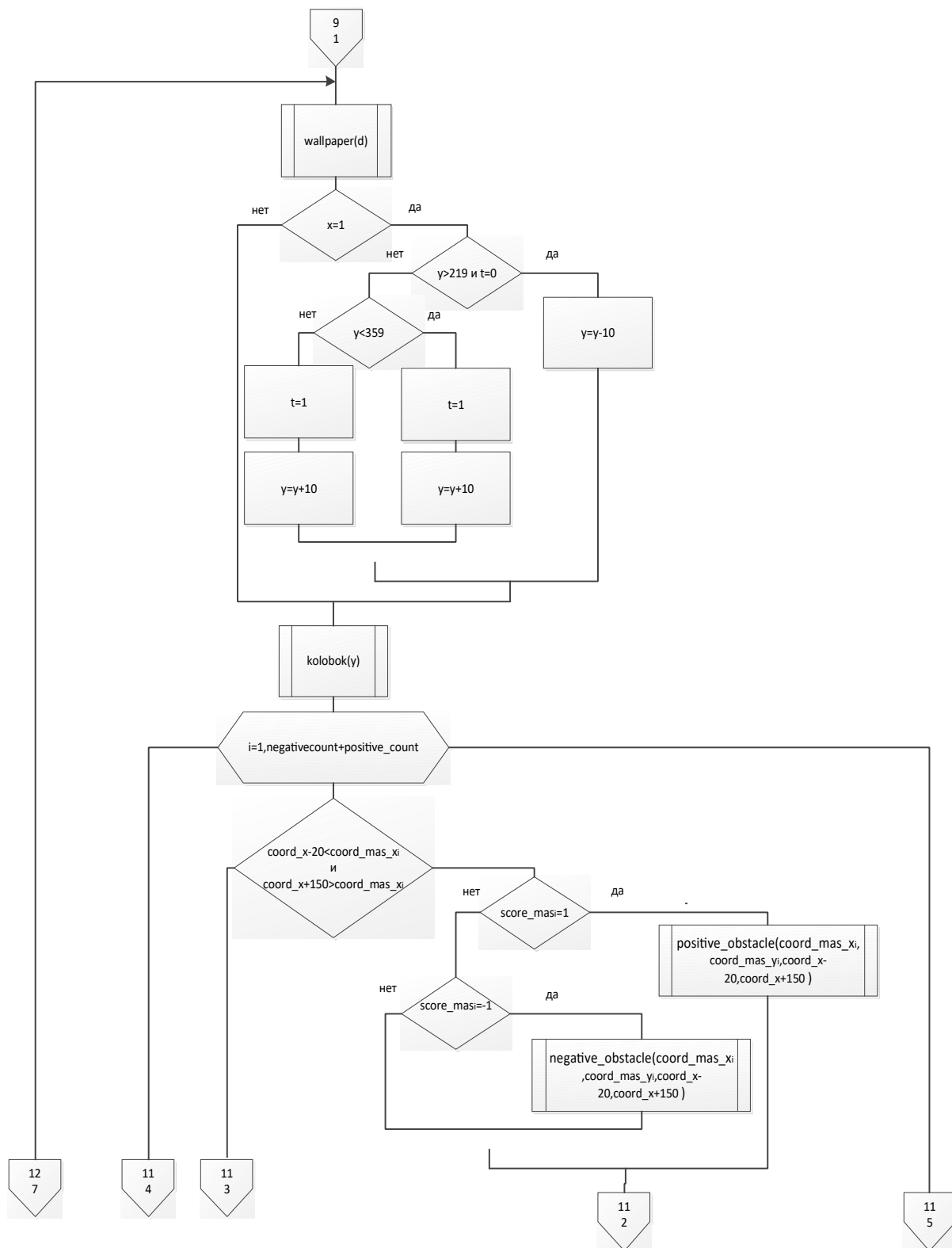


Рисунок 9 Процедура игры

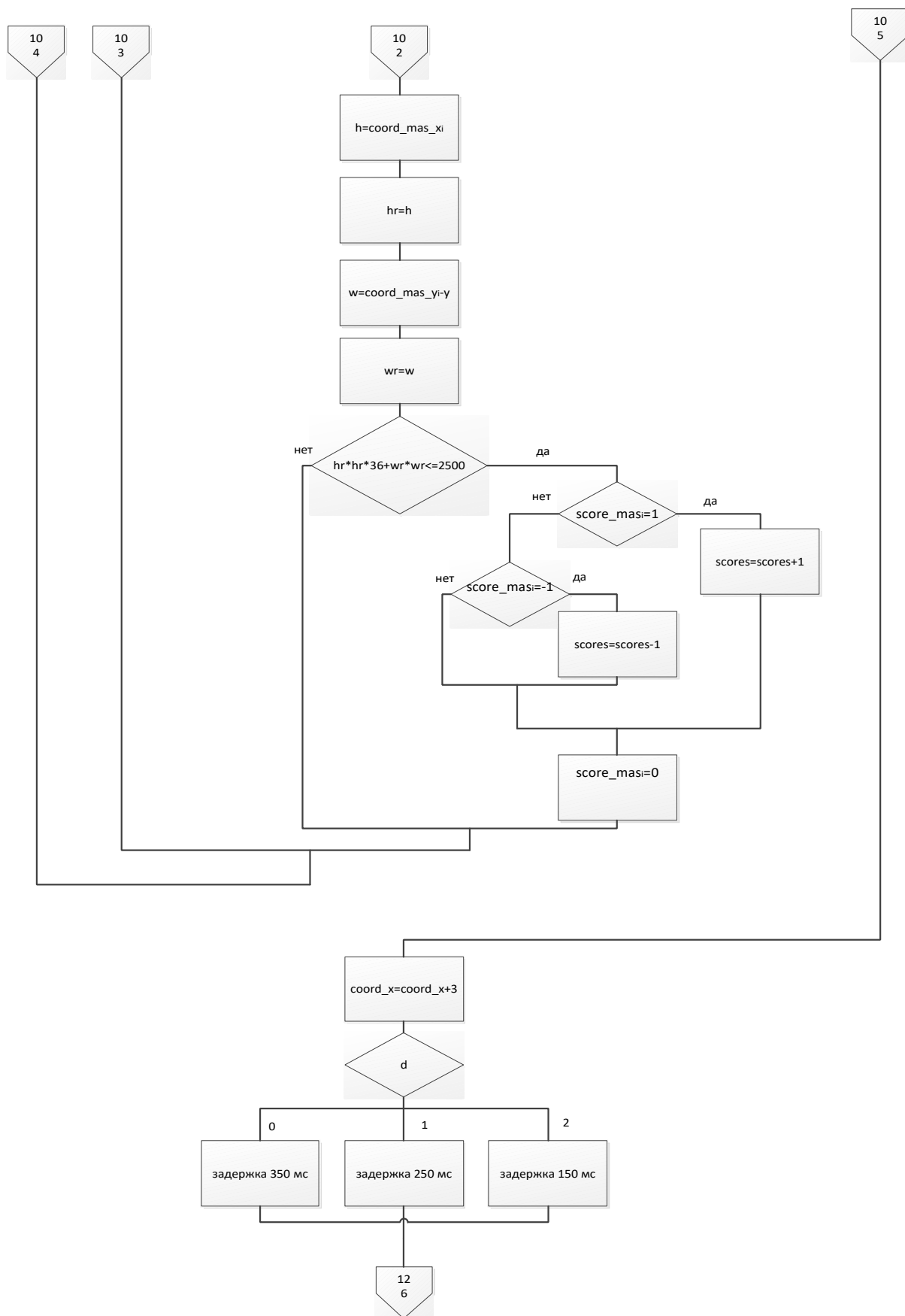


Рисунок 10 Процедура игры





**Рисунок 12 Процедура, показывающая инструкции для игры**

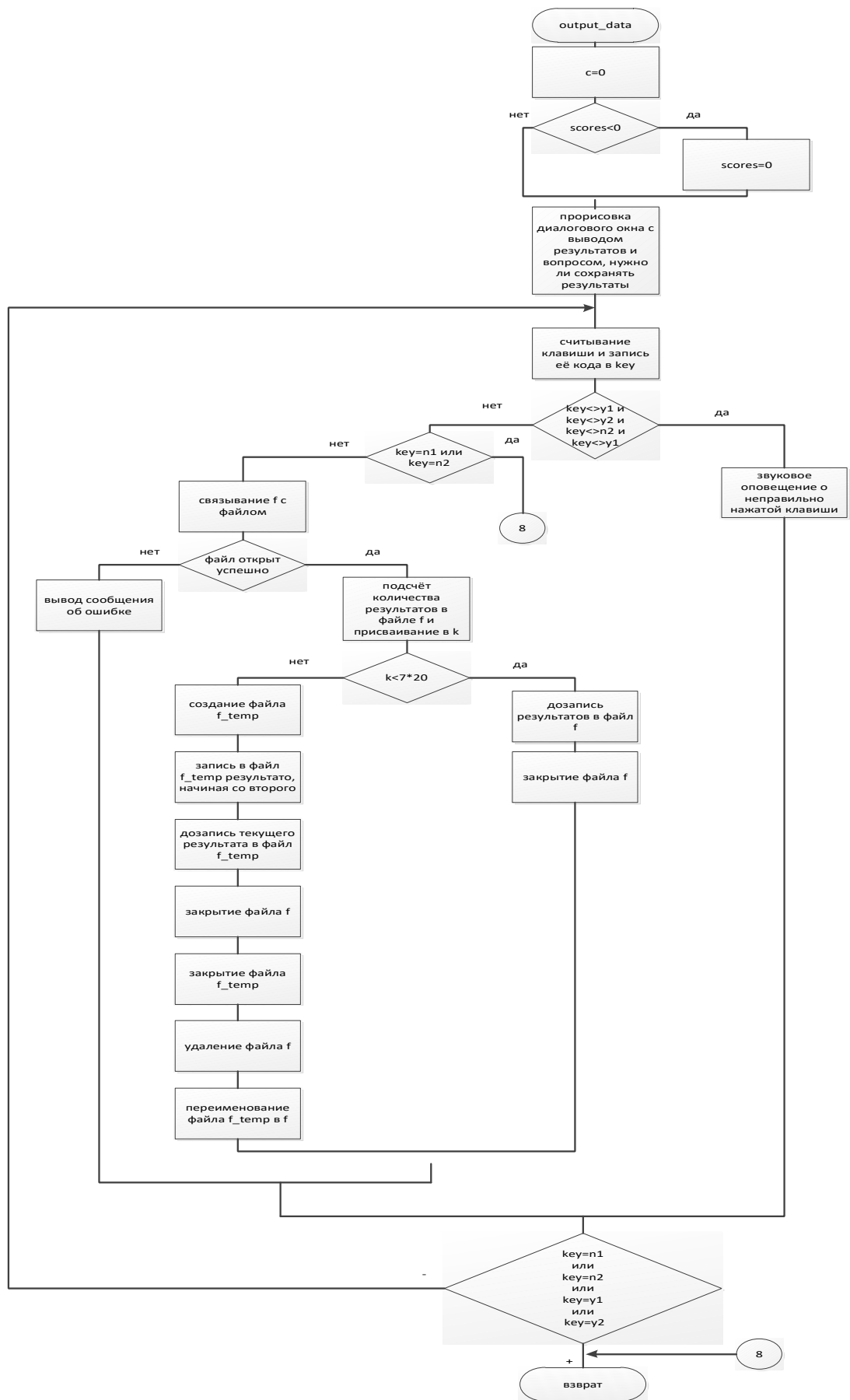


Рисунок 13 Процедура вывод результатов на экран и сохранения их в файл

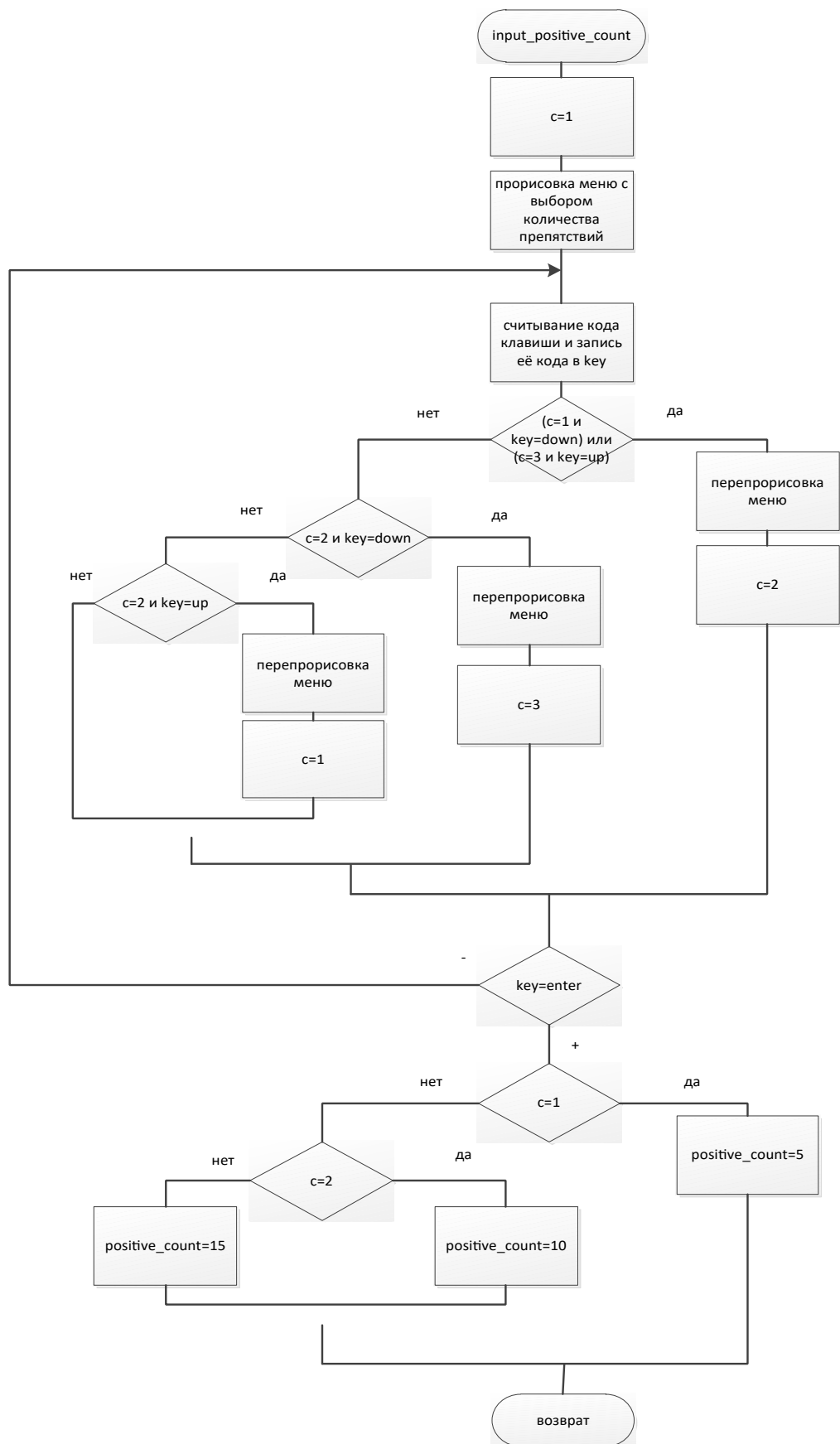


Рисунок 14 Процедура, записывающая количество «положительных» препятствий

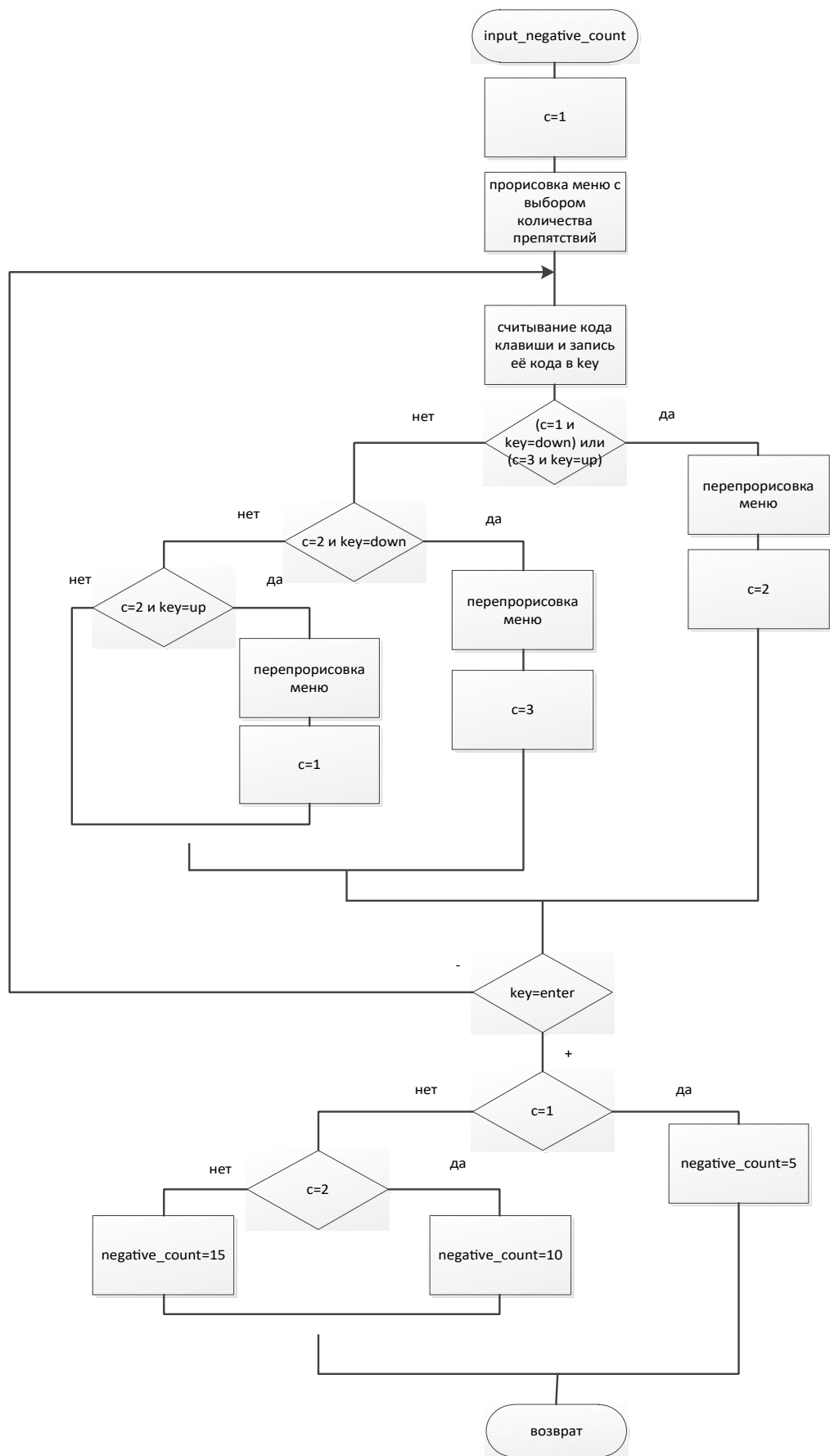


Рисунок 15 Процедура, записывающая количество «отрицательных» препятствий





**Рисунок 16 Процедура, заполняющая массив очков и массив препятствий**



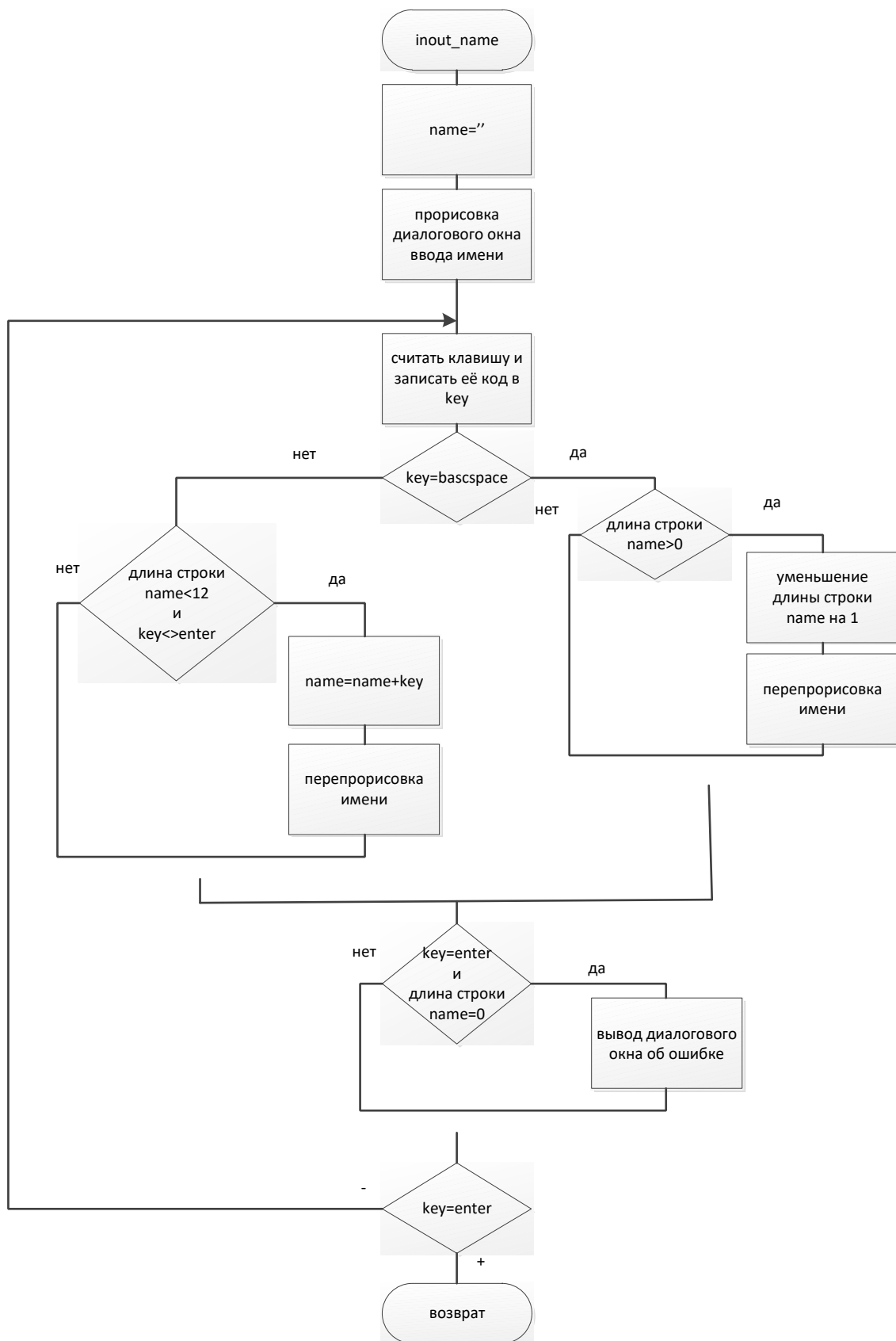
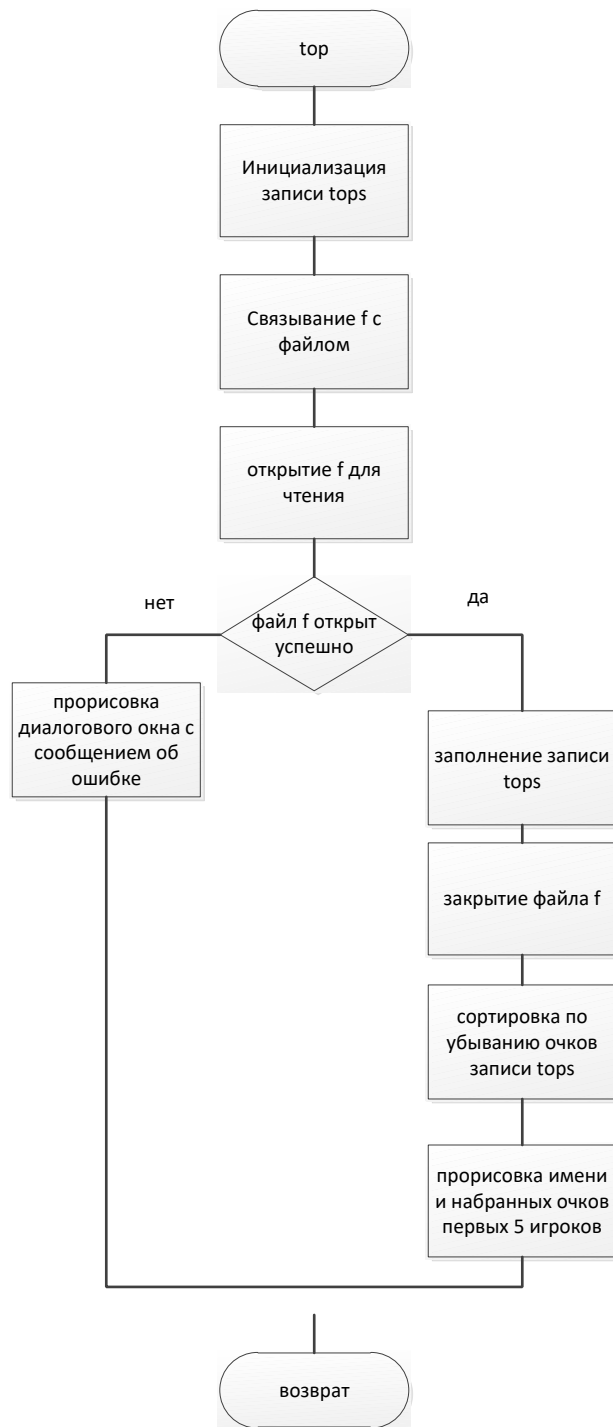
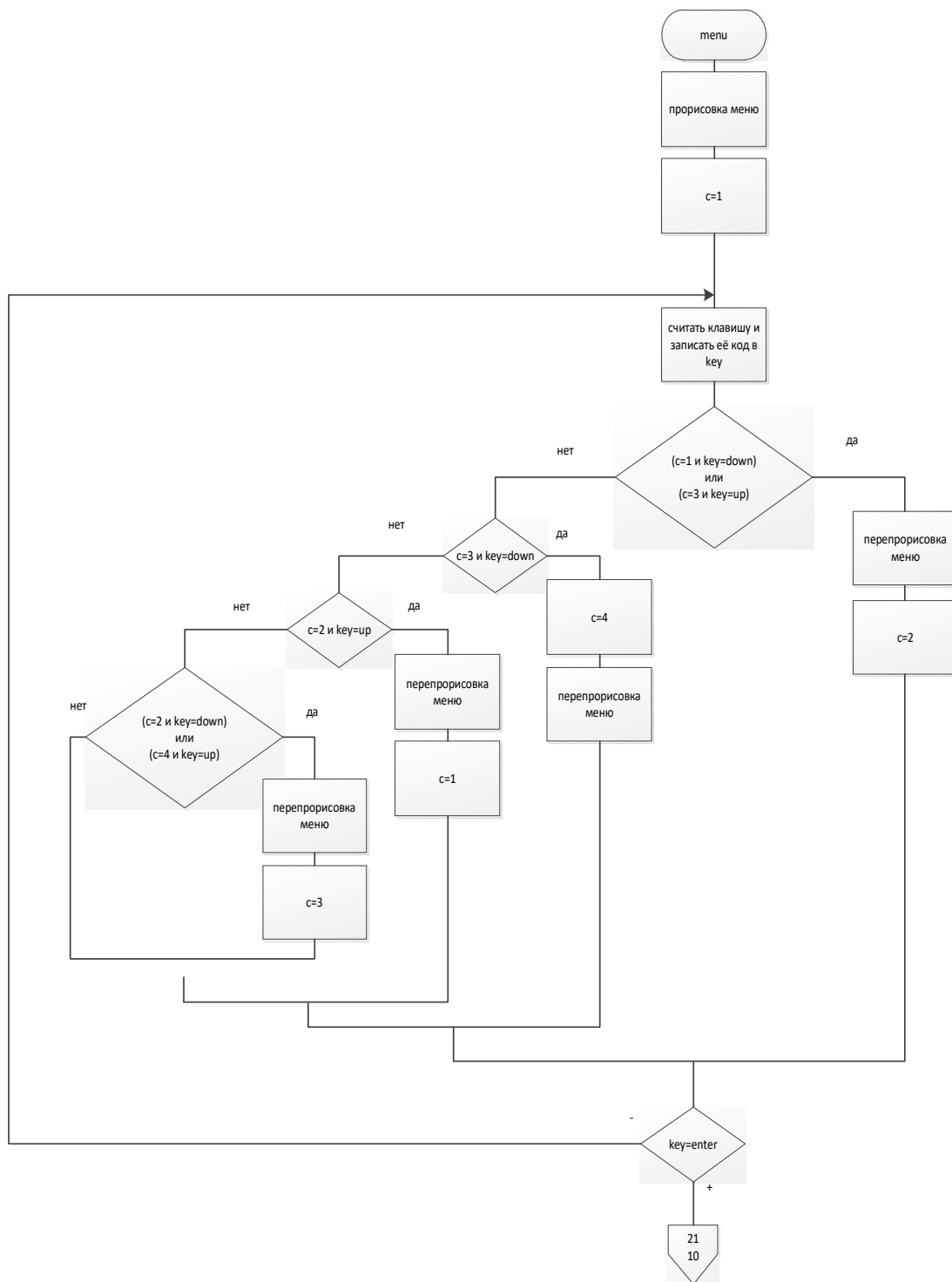


Рисунок 18 Процедура заполнения имени пользователя



**Рисунок 19 Процедура вывода топ-5 игроков**



**Рисунок 20 Процедура прорисовки меню**

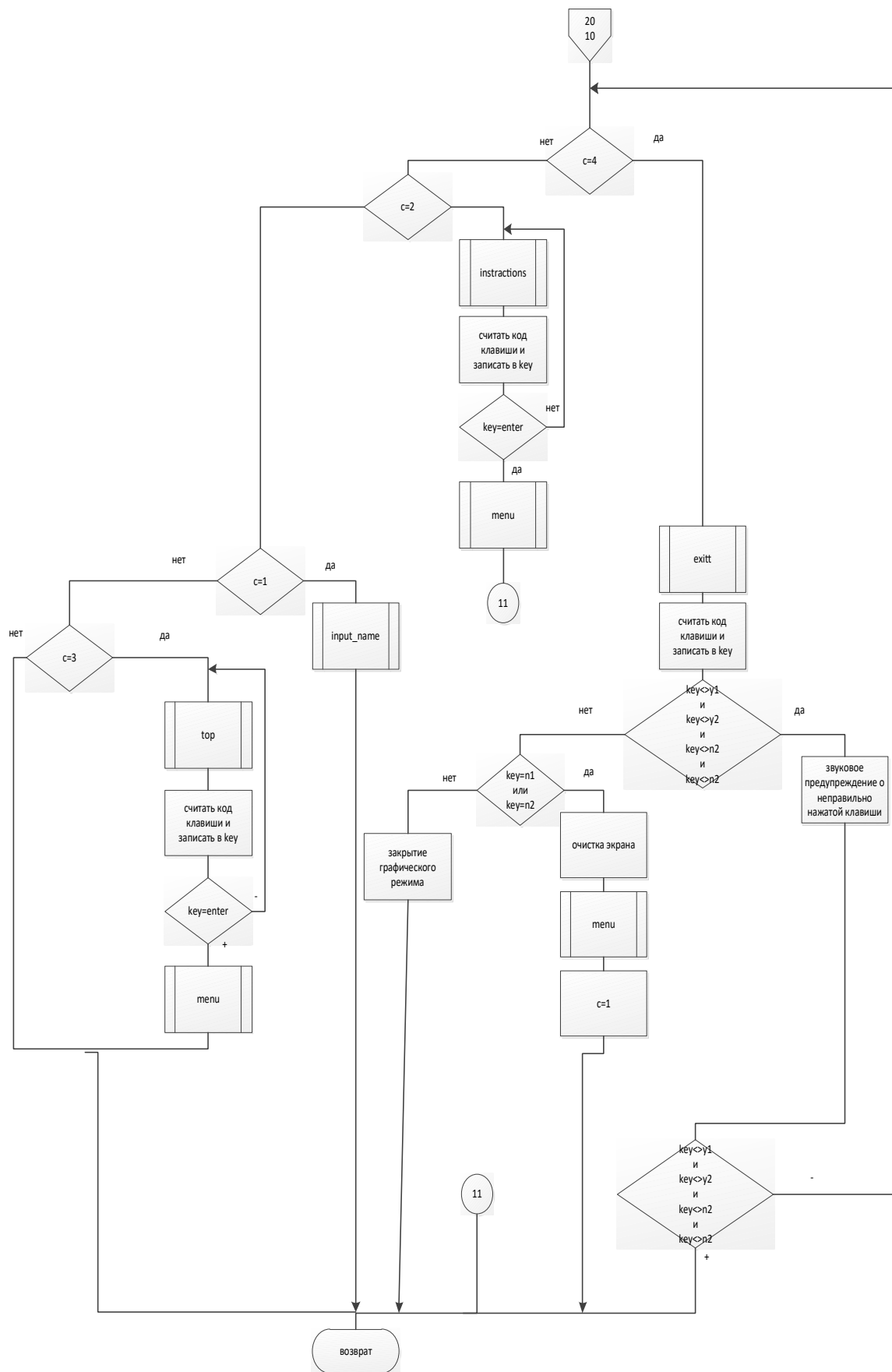


Рисунок 21 Процедура прорисовки меню

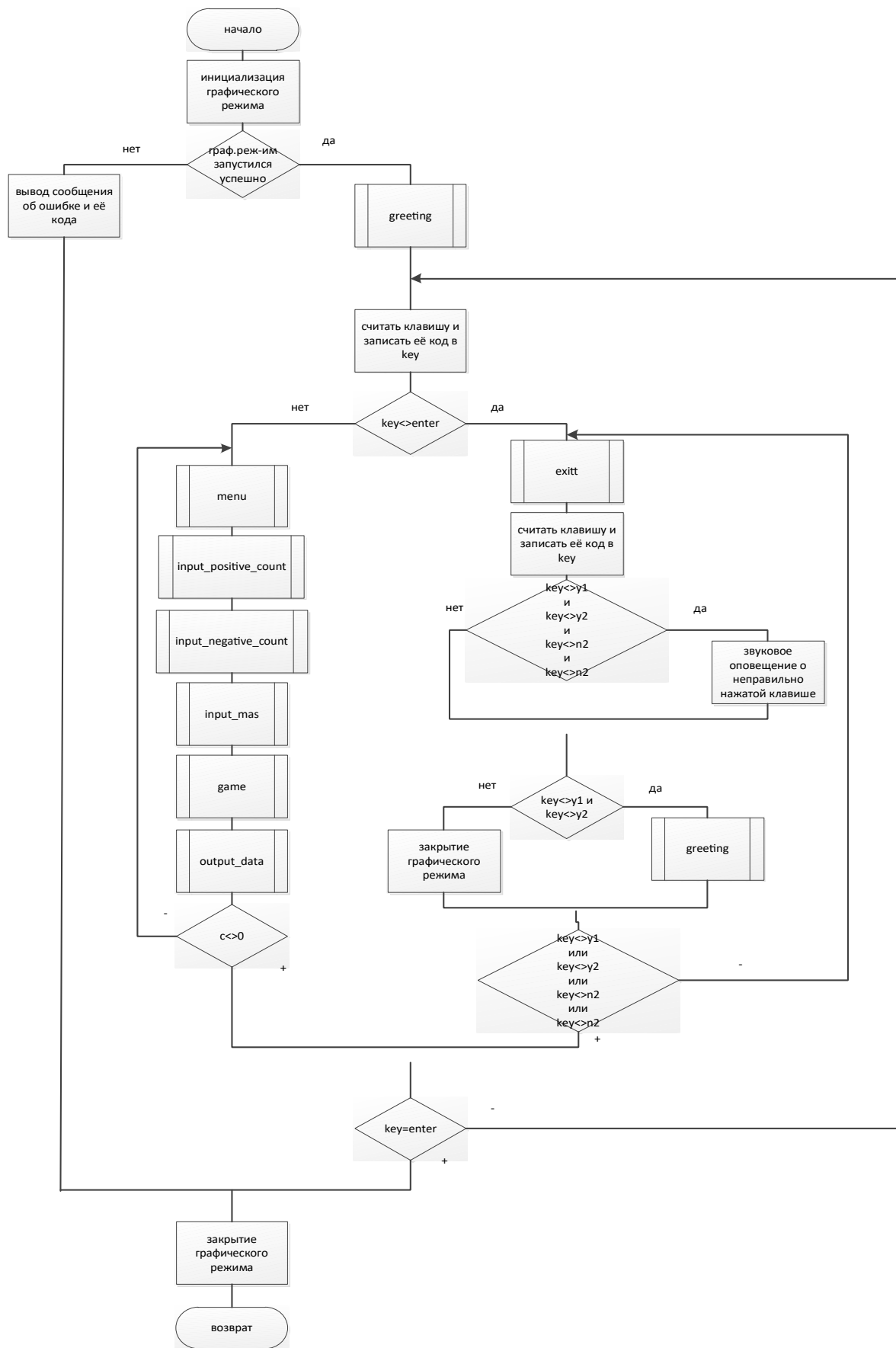


Рисунок 22 Основная программа

## 2.3. Описание структур, типов данных и глобальных переменных

### ■ Константы

- detect=0 – автоматическое распознавание типа видеоадаптера;
- enter=#13 – код клавиши «enter»;
- y1=#121 – код строчной клавиши «y»;
- y2=#89 – код прописной клавиши «Y»;
- n1=#78 – код прописной клавиши «N»;
- n2=#110 – код строчной клавиши «n»;
- upw1=#87 – код прописной клавиши «W»;
- upw2=#119 – код строчной клавиши «w»;
- right1=#68 – код прописной клавиши «D»;
- right2=#100 – код строчной клавиши «d»;
- left1=#65 – код прописной клавиши «A»;
- left2=#97 – код строчной клавиши «a»;
- esc=#27 – код клавиши «esc»;
- backspace=#8 – код клавиши «backspace»;

### ■ Тип

- mas=array [1..30] of integer – массив из 30 целочисленных элементов;

### ■ Переменные

- gd, gm, errorcode: integer – переменные для работы с графическим режимом;
- scores: integer – набранные очки пользователя;
- prov: integer – координата конца пути колобка;
- negative\_count: byte – количество «отрицательных» препятствий;
- positive\_count: byte – количество «положительных» препятствий;
- c: byte – вспомогательная переменная при работе с меню;
- key: код нажатой клавиши;
- score\_mas: mas – массив, определяющий тип следующего препятствия – «положительное» или «отрицательное»;
- coord\_mas\_x: mas – массив с координатами X препятствия;
- coord\_mas\_y: mas – массив с координатами Y препятствия;



- Процедуры
  - greeting – приветствие пользователя;
  - exitt – создание диалогового окна подтверждения выхода из игры;
  - kolobok – прорисовка колобка ;
  - negative\_obstacle – прорисовка «отрицательного» препятствия;
  - positive\_obstacle – прорисовка «положительного» препятствия ;
  - wallpaper – прорисовка неба, на фоне которого движется колобок, и некоторой технической информации;
  - game – процедура игрового процесса;
  - output\_data – вывод результатов на экран и сохранение их в файл;
  - instructions – выводит инструкции игры для пользователя;
  - input\_positive\_count – ввод количества «положительных» препятствий;
  - input\_negative\_count – ввод количества «отрицательных» препятствий;
  - input\_mas – генерация координат препятствий, а также генерация их очередности появления
  - input\_name – ввод имени пользователя
  - top – вывод на экран топ-5 игроков
  - menu – процедура меню

## 2.4. Описание процедур и функций

Работа основного модуля начинается инициализации графического режима. Если обнаружена ошибка, то на экран выводится сообщение о ней и происходит выход из программы.

### 1. Procedure greeting; (рис.2, стр.6)

- Входных данных нет
- Локальных переменных нет
- Выходные данные:
  - Изображение приветствия

С помощью стандартных процедур рисуется приветствие пользователя с предложением нажать клавишу «enter» для продолжения и любую другую для выхода из игры.

## 2. Procedure exitt; (рис.3, стр.7)

- Входных данных нет
- Локальных переменных нет
- Выходные данные:
  - Изображение диалогового окна подтверждения выхода из игры

С помощью стандартных процедур рисуется диалоговое окно подтверждения выхода из игры. При нажатии клавиши «n» будет произведён возврат на приветствие. При нажатии клавиши «y» произойдет выход из игры и закрытие программы. При нажатии любой другой клавиши — звуковое оповещение о неправильно нажатой клавише.

## 3. Procedure kolobok(const y:word); (рис.4, стр.7)

- Входные данные:
  - у-координата «у» центра колобка;
- Локальных переменных нет
- Выходные данные:
  - Изображение колобка

С помощью стандартных процедур рисуется колобок с центром в точке (180,y) и радиусом 40 пикселей

## 4. Procedure negative\_obstacle(const x:integer; const y:integer; const left:integer; const right:integer); (рис.5, стр.7)

- Входные данные:
  - x – мировая координата «x» препятствия;
  - y – координата «у» препятствия в пикселях;
  - left – мировая координата левого конца видимой области;
  - right – мировая координата правого конца видимой области;
- Локальные переменные:
  - k: real – хранит расположение препятствия относительно краёв видимой области по координате «x»;
  - paint\_x: integer – округляет k до ближайшего целого для дальнейшего использования в стандартных процедурах;
- Выходные данные:

- Изображение «отрицательного» препятствия

Высчитывается расположение «отрицательного» препятствия относительно краёв видимой области по координате «х», это отношение приводится к целочисленному виду, путем округления до ближайшего целого. Далее происходит прорисовка «отрицательного» препятствия в точке (paint\_x,y) и радиусом 10 пикселей.

5. Procedure positive\_obstacle(const x:integer; const y:integer; const left:integer; const right:integer); (рис.6, стр.8)

- Входные данные:
  - x – мировая координата «х» препятствия;
  - y – координата «у» препятствия в пикселях;
  - left – мировая координата левого конца видимой области;
  - right – мировая координата правого конца видимой области;
- Локальные переменные:
  - k: real – хранит расположение препятствия относительно краёв видимой области по координате «х»;
  - paint\_x: integer – округляет k до ближайшего целого для дальнейшего использования в стандартных процедурах;
- Выходные данные:
  - Изображение «положительного» препятствия

Высчитывается расположение «положительного» препятствия относительно краёв видимой области по координате «х», это отношение приводится к целочисленному виду, путем округления до ближайшего целого. Далее происходит прорисовка «положительного» препятствия в точке (paint\_x,y) и радиусом 10 пикселей.

6. Procedure wallpaper(const d:byte); (рис.7, стр.8)

- Входные данные
  - d – переменная «скорости» коlobка, принимает 3 значения: 0 – минимальная, 1 – средняя, 2 – максимальная;
- Локальных переменных нет
- Выходные данные:
  - изображение неба, на фоне которого движется коlobок, а также изображение технической информации и подсказок;

С помощью стандартных процедур выполняется перепрорисовка неба, для удаления препятствий. Также на экран выводятся подсказки об управляющих клавишах, и в зависимости от  $d$  рисуется индикатор скорости колобка.

#### 7. Procedure game; (рис.8-рис.11, стр.9-стр.12)

- Входных данных нет
- Локальные переменные
  - $d, t, x: \text{byte}$  – вспомогательные переменные
  - $i, j: \text{integer}$  – вспомогательные переменные
  - $hr, hw: \text{real}$  – используются для приведения типов
  - $h: \text{longint}$  – вспомогательная для проверки, съел ли колобок препятствие или нет по оси  $Ox$
  - $w: \text{integer}$  – вспомогательная для проверки, съел ли колобок препятствие или нет по оси  $Oy$
  - $coord\_x$  – мировая координата колобка по оси  $Ox$
  - $y$  – координата колобка по оси  $Oy$  в пикселях
- Глобальные переменные
  - $scores: \text{integer}$  – количество очков, набранных пользователем
- Выходные данные
  - игровой процесс

Происходит отрисовка игрового поля. Затем начинается движение колобка, который встречает на своём пути ранее сгенерированные препятствия. Если колобок задевает «положительное» препятствие, то количество очков увеличивается на 1, иначе уменьшается на 1. При нажатии клавиши «w» колобок будет совершать прыжок. При нажатии клавиши «d» колобок будет увеличивать скорость при нажатии «a» - уменьшать. Достигается это изменением времени задержки. После прохождения всех препятствий появится диалоговое окно с выводом количества очков, а также с вопросом, нужно ли сохранять результаты.

#### 8. Procedure output\_data; (рис.13, стр.13)

- Входные данные отсутствуют
- Локальные переменные:

- h,m,s,hund:word – вспомогательные переменные для получения даты и времени;
- temp: string – вспомогательная переменная для «красивого» вывода даты и времени в файл;
- strin: string – вспомогательная переменная для считывания строк из файла;
- i:byte – переменная счётчик;
- f: text – файловая переменная, связывается с файлом, в котором хранятся результаты;
- f\_temp: text – вспомогательная файловая переменная для создания временного файла с результатами;
- Глобальные переменные:
  - scores: integer – подсчёт количества набранных очков пользователя;
  - c:byte – вспомогательная переменная для работы с меню;
  - key:char – код нажатой клавиши
- Выходные данные
  - файл с результатами игр;
  - диалоговое окно с результатами данной игры;

Если пользователь набрал отрицательное количество очков, то количество очков становится равным 0. Далее выводится диалоговое окно с результатами и предложением их сохранить в файл. При нажатии клавиши «n» (не сохранять) пользователь вернётся в главное меню. При нажатии клавиши «y» (сохранять) возможны 2 варианта: если файла с результатами не существует, то будет выдано сообщение об ошибке с просьбой нажать клавишу «enter» для продолжения. Если файл существует и успешно открыт, то после нажатия клавиши «y» пользователь автоматически перейдёт в меню и произойдёт сохранение результатов в файл. Если количество результатов на момент сохранения будет 20 и более, то произойдёт удаление одной старой записи и добавление новой (через дополнительный временный файл).

#### 9. Procedure instructions; (рис.12, стр.12)

- Входных данных нет
- Локальных переменных нет
- Выходные данные

- инструкции по игре

С помощью стандартных процедур выполняется прорисовка инструкций для игры.

#### 10.Procedure input\_positive\_count; (рис.14, стр.14)

- Входных данных нет
- Глобальные переменные
  - c: byte - вспомогательная переменная для работы с меню;
  - key: char – код нажатой клавиши
  - positive\_count: byte – количество «положительных» препятствий
- Локальных переменных нет
- Выходные данные
  - количество «положительных» препятствий

С помощью стандартных процедур рисуется меню выбора количества «положительных» препятствий. Пользователь выбирает количество, которое затем и записывается в positive\_count.

#### 11.Procedure input\_negative\_count; (рис.15,стр.15)

- Входных данных нет
- Глобальные переменные
  - c: byte - вспомогательная переменная для работы с меню;
  - key: char – код нажатой клавиши
  - negative\_count: byte – количество «положительных» препятствий
- Локальных переменных нет
- Выходные данные
  - количество «отрицательных» препятствий

С помощью стандартных процедур рисуется меню выбора количества «отрицательных» препятствий. Пользователь выбирает количество, которое затем и записывается в negative\_count.

#### 12.Procedure input\_mas; (рис.16-рис.17, стр.16-стр.17)

- Входных данных нет
- Глобальные переменные
  - negative\_count: byte – количество «отрицательных» препятствий;

- positive\_count: byte – количество «положительных» препятствий;
- coord\_mas\_x: mas – массив с координатами X препятствия;
- coord\_mas\_y: mas – массив с координатами Y препятствия;
- prov: integer – координата конца пути коlobка;
- Локальные переменные:
  - i,j,k,l: integer – вспомогательные переменные
  - buffer\_mas:mas - временный массив
  - temp: word – временная переменная
- Выходные данные:
  - массив с координатами коlobка по оси Oy
  - массив с координатами коlobка по оси Ox
  - массив, определяющий тип следующего препятствия – «положительное» или «отрицательное»;

Временный массив buffer\_mas заполняется сначала -1 (количество зависит от глобальной переменной negative\_count), затем 1 (количество зависит от глобальной переменной positive\_count). Далее происходит генерация случайным образом мировых координат коlobка по оси Ox в определенном диапазоне, зависящим от общего количества препятствий. Затем происходит генерация случайным образом координат коlobка (в пикселях) по оси Oy. Следующим шагом происходит сортировка по возрастанию массива с координатами по оси Ox. В конце формируем массив, определяющий тип следующего препятствия рандомно, используя временный массив buffer\_mas.

### 13.Procedure input\_name; (рис.18, стр.18)

- Входных данных нет
- Локальных переменных нет
- Выходные данные
  - Имя пользователя

Прорисовывается экран ввода имени пользователя. Для подтверждения ввода имени необходимо нажать клавишу «enter». При попытке ввести пустое имя появится окно с ошибкой. При нажатии клавиши «backspace» произойдет удаление последнего

символа. При нажатии клавиши «enter» имя будет сохранено и пользователь перейдёт на следующий шаг.

#### 14. Procedure top (рис.19, стр.19)

- Входных данных нет
- Типы
  - tops – запись, содержащая 2 поля: имя и количество очков
  - mas=array[1..20] of tops - массив записей
- Локальные переменные
  - f: text – файловая переменная, связывается с файлом, в котором хранятся результаты;
  - a:mas – переменная записей
  - temp,temp\_name,s: string - вспомогательные переменные
  - i: byte – вспомогательная переменная
  - j,t,code: integer – вспомогательные переменные
- Выходные данные
  - Топ-5 игроков

Для начала происходит связывание с файлом. Если файл не существует или не открыт успешно, появится диалоговое окно с ошибкой и пользователь перейдет обратно в меню. При успешном открытии начнётся заполнение записи. Затем запись будет отсортирована по убыванию количества очков и первые 5 пользователей будут выведены на экран.

#### 15. Procedure menu (рис.20-рис.21, стр.20-стр.21)

- Входных данных нет
- Локальные переменные
  - c:byte – вспомогательная переменная для работы с меню
- Выходные данные
  - Меню

С помощью стандартных процедур происходит прорисовка меню. При нажатии клавиши «вверх» или «вниз» меню перепрорисовывается. При нажатии клавиши «enter» пользователь попадает в необходимый раздел.





## **Заключение**

В ходе выполнения данной курсовой работы была написана программа - «Колобок». Разработанное средство получилось нетребовательным к системным ресурсам, однако из-за свойств среды приложение работает с торможением и создаётся впечатление о том, что игра плохо проработана. Однако это не так. Также в игре прекрасно проработан диалог с пользователем, исключены тупиковые ситуации. Можно отметить, что в ходе выполнения курсовой работы мною более глубоко был изучен язык программирования «Pascal», его графические возможности, а также был приобретен опыт в создании графического интерфейса.

## Список литературы

1. Алексеев Е.Р. Free Pascal и Lazarus [Электронный ресурс] : учеб. по программированию / Е. Р. Алексеев, О. В. Чеснокова, Т. В. Кучер. - Электрон. текстовые дан. - Москва : ДМК Пресс, 2010. Режим доступа: <https://elib.bstu.ru/Reader/Book/7255>.
2. Стативко Р.У. Языки программирования [Электронный ресурс] : учеб. пособие / Р.У. Стативко, Е. А. Лазебная ; БГТУ им. В. Г. Шухова. - Электрон. текстовые дан. - Белгород: Изд-во БГТУ им. В. Г. Шухова, 2015. Режим доступа:
3. Зюзьков В.М. Программирование: Курс лекций – Томск, 1999.

## Приложение 1 Листинг программы

```
uses
    graph,
    dos,
    crt;

type
    mas=array [1..30] of integer;

const
    detect=0;
    enter=#13;
    y1=#121;
    y2=#89;
    n1=#78;
    n2=#110;
    up=#72;
    upw1=#87;
    upw2=#119;
    down=#80;
    right1=#68;
    right2=#100;
    left1=#65;
    left2=#97;
    esc=#27;
    backspace=#8;

var
    gd,gm,errorcode,prov,scores:integer;
    key:char;
    name:string;
    negative_count,positive_count,c:byte;
    score_mas,coord_mas_x,coord_mas_y:mas;

procedure
```

```

greeting;
begin
    setbkcolor(green);
    setcolor(yellow);
    settextstyle(0,0,10);
    outtextxy(50,50,'КОЛОБОК');
    setcolor(white);
    settextstyle(7,0,4);
    outtextxy(70,200,'Здравствуй, дорогой друг!');
    outtextxy(20,250,'Готов к новым приключениям?');
    outtextxy(70,300,'Если да - жми ENTER!');
    outtextxy(5,350,'Иначе - любую другую клавишу');
end;

procedure
    exitt;
begin
    setfillstyle(1,8);
    bar(50,150,600,400);
    setcolor(white);
    settextstyle(0,0,2);
    outtextxy(95,175,'Вы уверены, что хотите выйти?');
    setfillstyle(1,white);
    bar(60,330,200,390);
    bar(450,330,590,390);
    setcolor(0);
    settextstyle(0,0,6);
    outtextxy(105,335,'Y');
    outtextxy(505,335,'N');
    setcolor(yellow);
    outtextxy(95,270,'Да');
    outtextxy(460,270,'Нет');
end;

procedure
    kolobok(const y:word);
begin
    setcolor(yellow);
    circle(180,y,40);
    setfillstyle(1,yellow);
    floodfill(180,y,yellow);
end;

procedure
    negative_obstacle(const x:integer; const y:integer; const left:integer;
const right:integer);
var
    paint_x:integer;
    k:real;
begin
    setfillstyle(1,red);
    setcolor(red);
    k:=(x-left)/(right-left)*640;
    paint_x:=round(k);
    circle(paint_x,y,10);
    floodfill(paint_x,y,red);

```

```

end;

procedure
  positive_obstacle(const x:integer; const y:integer; const left:integer;
const right:integer);
var
  paint_x:integer;
  k:real;
begin
  setcolor(yellow);
  setfillstyle(1,yellow);
  k:=(x-left)/(right-left)*640;
  paint_x:=round(k);
  circle(paint_x,y,10);
  floodfill(paint_x,y,yellow);
end;

procedure
  wallpaper(const d:byte);
begin
  setcolor(blue);
  setfillstyle(1,blue);
  bar(0,0,650,400);
  floodfill(5,100,blue);
  settextstyle(7,0,4);
  setcolor(white);
  outtextxy(10,10,'Скорость');
  outtextxy(450,10,'Прыжок');
  outtextxy(520,50,'W');
  outtextxy(170,50,'A');
  outtextxy(310,50,'D');
  line(200,30,300,30);
  case d of
    0:
      begin
        setcolor(red);
        line(200,0,200,60);
        setcolor(blue);
        line(250,0,250,60);
        line(300,0,300,60);
      end;
    1:
      begin
        setcolor(red);
        line(250,0,250,60);
        setcolor(blue);
        line(200,0,200,60);
        line(300,0,300,60);
      end;
    2:
      begin
        setcolor(red);
        line(300,0,300,60);
        setcolor(blue);
        line(250,0,250,60);
        line(200,0,200,60);
      end;
  end;
end;

```

```

        end;
    end;

procedure
game;
var
    d,t,x:byte;
    coord_x,w,y,i,j:integer;
    hr,wr:real;
    h:longint;
begin
    coord_x:=30;
    setlinestyle(0,0,3);
    line(0,400,700,400);
    setfillstyle(1,green);
    floodfill(5,410,blue);
    d:=0;
    x:=0;
    t:=0;
    i:=1;
    y:=359;
    scores:=0;
    repeat
        wallpaper(d);
        if (x=1)
            then
                if ((y>219) and (t=0))
                    then
                        y:=y-10
                    else
                        if (y<359)
                            then
                                begin
                                    t:=1;
                                    y:=y+10;
                                end
                            else
                                if (y>=359)
                                    then
                                        begin
                                            x:=0;
                                            t:=0;
                                        end;
                                else
                                    kolobok(y);
                                    for i:=1 to negative_count+positive_count do
                                        begin
                                            if (((coord_x-20)<coord_mas_x[i]) and
                                                ((coord_x+150)>coord_mas_x[i]))
                                                then
                                                    begin
                                                        if score_mas[i]=1
                                                            then
                                                                positive_obstacle(coord_mas_x[i],coord_mas_y[i],coord_x-20,coord_x+150);
                                                                if score_mas[i]=-1
                                                                    then
                                                                        negative_obstacle(coord_mas_x[i],coord_mas_y[i],coord_x-20,coord_x+150);
                                                                h:=coord_mas_x[i]-coord_x-25;

```

```

        hr:=h;
        w:=(coord_mas_y[i]-y);
        wr:=w;
        if ((hr*hr*36+wr*wr)<=2500)
            then
                begin
                    if (score_mas[i]=1)
                        then
                            scores:=scores+1
                        else
                            if (score_mas[i]=-1)
                                then
                                    scores:=scores-1;
                                score_mas[i]:=0;
                            end;
                end;
            end;
        end;
        coord_x:=coord_x+3;
        case d of
            0:
                delay(350);
            1:
                delay(250);
            2:
                delay(150);
        end;
        while (keypressed) do
            begin
                key:=readkey;
                if ((key=right1) or (key=right2))
                    then
                        case d of
                            0:
                                d:=1;
                            1:
                                d:=2;
                        end
                    else
                        if ((key=left1) or (key=left2))
                            then
                                case d of
                                    1:
                                        d:=0;
                                    2:
                                        d:=1;
                                end
                            else
                                if ((key=upw1) or (key=upw2))
                                    then
                                        x:=1;
                                    end;
                                end;
                        until (coord_x>prov);
                    end;

```

```

procedure
    output_data;
    var
        h,m,s,hund,k:word;
        temp,strin:string;

```

```

i:byte;
f,f_temp:text;
begin
  c:=0;
  if (scores<0)
    then
      scores:=0;
  setfillstyle(1,8);
  bar(50,150,600,400);
  setcolor(white);
  settextstyle(0,0,2);
  outtextxy(150,175,'Ваш счёт:');
  str(scores,temp);
  outtextxy(300,175,temp);
  outtextxy(150,200,'Сохранить результаты?');
  setfillstyle(1,white);
  bar(60,330,200,390);
  bar(450,330,590,390);
  setcolor(green);
  settextstyle(0,0,6);
  outtextxy(105,335,'Y');
  outtextxy(505,335,'N');
  setcolor(yellow);
  outtextxy(95,270,'Да');
  outtextxy(460,270,'Нет');
  repeat
    key:=readkey;
    if ((key<>y1) and (key<>y2) and (key<>n1) and (key<>n2) and
(key<>y1) and (key<>y2))
      then
        begin
          sound(400);
          delay(150);
          nosound;
        end
      else
        if ((key=n1) or (key=n2))
          then
            exit
          else
            begin
              k:=0;
              assign(f,'y:/kolobok/results.txt');
              {$i-};
              reset(f);
              {$i+};
              if ioresult<>0
                then
                  begin
                    setfillstyle(1,8);
                    bar(50,150,600,400);
                    setcolor(white);
                    settextstyle(0,0,2);
                    outtextxy(70,175,'Произошла ошибка');
                    outtextxy(70,200,'Результаты не будут
сохранены');

                    setcolor(yellow);
                    settextstyle(0,0,6);
                    outtextxy(270,300,'OK');
                    repeat

```



```

        key:=readkey;
        if (key<>'enter')
            then
                begin
                    sound(400);
                    delay(150);
                    nosound;
                end
            else
                exit;
            until (key='enter')
        end
    else
        begin
            while not eof(f) do
                begin
                    readln(f, strin);
                    k:=k+1;
                end ;
            close(f);
            if (k<(7*20))
                then
                    begin
                        append(f);
                        temp:='';
                        gettime(h,m,s,hund);
                        str(m,temp);
                        if (length(temp)=1)
                            then
                                temp:='0'+temp;
                        write(f, 'User: ');
                        writeln(f, name);
                        write(f, 'Time: ');
                        write(f, h);
                        write(f, ':');
                        writeln(f, temp);
                        write(f, 'Date: ');
                        getdate(h,m,s,hund);
                        write(f, s);
                        write(f, '/');
                        str(m,temp);
                        if (length(temp)=1)
                            then
                                temp:='0'+temp;
                        write(f, temp);
                        write(f, '/');
                        writeln(f, h);
                        write(f, 'Number of positive ob-
stacles: ');
                        writeln(f, positive_count);
                        write(f, 'Number of negative ob-
stacles: ');
                        writeln(f, negative_count);
                        write(f, 'Score: ');
                        writeln(f, scores);
                        writeln(f, '----- ');
                        close(f);
                    end
                else

```

```

selts_temp.txt');

begin
    assign(f_temp,'y:/kolobok/re-

    rewrite(f_temp);
    reset(f);
    for i:=1 to 7 do
        readln(f,strin);
    while not eof(f) do
        begin
            readln(f,strin);
            writeln(f_temp,strin);
        end;
    temp:='';
    gettime(h,m,s,hund);
    str(m,temp);
    if (length(temp)=1)
        then
            temp:='0'+temp;
    write(f_temp,'User: ');
    writeln(f_temp,name);
    write(f_temp,'Time: ');
    write(f_temp,h);
    write(f_temp,':');
    writeln(f_temp,temp);
    write(f_temp,'Date: ');
    getdate(h,m,s,hund);
    write(f_temp,s);
    write(f_temp,'/');
    str(m,temp);
    if (length(temp)=1)
        then
            temp:='0'+temp;
    write(f_temp,temp);
    write(f_temp,'/');
    writeln(f_temp,h);
    write(f_temp,'Number of positive

    writeln(f_temp,positive_count);
    write(f_temp,'Number of negative

    writeln(f_temp,negative_count);
    write(f_temp,'Score: ');
    writeln(f_temp,scores);

writeln(f_temp,'----- ');

    close(f);
    close(f_temp);
    erase(f);
    rename(f_temp,'y:/kolobok/re-

sults.txt');

end;

end;

until ((key=n1) or (key=n2) or (key=y1) or (key=y2))
end;

procedure
instructions;
begin

```

```

cleardevice;
setbkcolor(green);
setcolor(yellow);
settextstyle(0,0,7);
outtextxy(40,50,'Инструкции');
settextstyle(0,0,5);
outtextxy(250,350,'Ясно');
setcolor(white);
setfillstyle(1,yellow);
bar(20,130,50,160);
settextstyle(7,0,4);
outtextxy(70,120,' -тесто,укрепляет силы');
outtextxy(120,150,'колобка; +1 очко');
setfillstyle(1,red);
bar(20,190,50,220);
outtextxy(70,180,' -опасные хищники,наносят');
outtextxy(120,210,'урон колобку; -1 очко');
setfillstyle(1,white);
bar(200,400,450,470);
settextstyle(0,0,6);
setcolor(brown);
outtextxy(210,415,'ENTER');
end;

```

```

procedure
input_positive_count;
begin
cleardevice;
setbkcolor(green);
setcolor(yellow);
settextstyle(7,0,6);
outtextxy(140,10,'Количество');
outtextxy(70,60,'положительных');
outtextxy(110,110,'препятствий');
setcolor(white);
line(0,170,700,170);
setcolor(yellow);
settextstyle(7,0,6);
outtextxy(300,210,'5');
c:=1;
setcolor(white);
outtextxy(280,280,'10');
outtextxy(280,350,'15');
repeat
key:=readkey;
if ((c=1) and (key=down)) or ((c=3) and (key=up))
then
begin
setcolor(yellow);
outtextxy(280,280,'10');
setcolor(white);
outtextxy(300,210,'5');
outtextxy(280,350,'15');
c:=2;
end
else
if ((c=2) and (key=down))
then
begin

```

```

        setcolor(yellow);
        outtextxy(280,350,'15');
        setcolor(white);
        outtextxy(280,280,'10');
        outtextxy(300,210,'5');
        c:=3;
    end
else
    if ((c=2) and (key=up))
    then
        begin
            setcolor(yellow);
            outtextxy(300,210,'5');
            setcolor(white);
            outtextxy(280,280,'10');
            outtextxy(280,350,'15');
            c:=1;
        end
until (key=enter);
if (c=1)
then
    positive_count:=5
else
    if (c=2)
    then
        positive_count:=10
    else
        positive_count:=15;
end;
end;
```

```

procedure
input_negative_count;
begin
    cleardevice;
    setbkcolor(green);
    setcolor(yellow);
    settextstyle(7,0,6);
    outtextxy(140,10,'Количество');
    outtextxy(70,60,'отрицательных');
    outtextxy(110,110,'препятствий');
    setcolor(white);
    line(0,170,700,170);
    setcolor(yellow);
    settextstyle(7,0,6);
    outtextxy(300,210,'5');
    c:=1;
    setcolor(white);
    outtextxy(280,280,'10');
    outtextxy(280,350,'15');
    repeat
        key:=readkey;
        if (((c=1) and (key=down)) or ((c=3) and (key=up)))
        then
            begin
                setcolor(yellow);
                outtextxy(280,280,'10');
                setcolor(white);
                outtextxy(300,210,'5');
                outtextxy(280,350,'15');
```

```

        c:=2;
    end
else
    if ((c=2) and (key=down))
    then
        begin
            setcolor(yellow);
            outtextxy(280,350,'15');
            setcolor(white);
            outtextxy(280,280,'10');
            outtextxy(300,210,'5');
            c:=3;
        end
    else
        if ((c=2) and (key=up))
        then
            begin
                setcolor(yellow);
                outtextxy(300,210,'5');
                setcolor(white);
                outtextxy(280,280,'10');
                outtextxy(280,350,'15');
                c:=1;
            end
        until (key=enter);
    if (c=1)
    then
        negative_count:=5
    else
        if (c=2)
        then
            negative_count:=10
        else
            negative_count:=15;
        end;
end;

```

```

procedure
input_mas;
var
    i,l,k,j:byte;
    temp:word;
    buffer_mas:mas;
begin
    randomize;
    for i:=1 to negative_count do
        buffer_mas[i]:=-1;
    for i:=(negative_count+1) to (negative_count+positive_count) do
        buffer_mas[i]:=1;
    l:=negative_count+positive_count;
    case l of
        10:
            begin
                for i:=1 to l do
                    coord_mas_x[i]:=150+random(500);
                prov:=670;
            end;
        15:
            begin
                for i:=1 to l do

```

```

        coord_mas_x[i]:=150+random(1000);
        prov:=1170;
    end;
20:
    begin
        for i:=1 to l do
            coord_mas_x[i]:=150+random(1500);
            prov:=1670;
        end;
25:
    begin
        for i:=1 to l do
            coord_mas_x[i]:=150+random(2000);
            prov:=2170;
        end;
30:
    begin
        for i:=1 to l do
            coord_mas_x[i]:=150+random(2500);
            prov:=2620;
        end;
    end;
end;
for i:=1 to l do
    coord_mas_y[i]:=200+random(150);
for i:=1 to l-1 do
    for j:=i+1 to l do
        if coord_mas_x[i]>coord_mas_x[j]
        then
            begin
                temp:=coord_mas_x[i];
                coord_mas_x[i]:=coord_mas_x[j];
                coord_mas_x[j]:=temp;
            end;
    end;
for i:=1 to l do
    begin
        k:=random(l)+1;
        score_mas[i]:=buffer_mas[k];
        for j:=k to (l-1) do
            buffer_mas[j]:=buffer_mas[j+1];
        l:=l-1;
    end;
end;
end;

```

```

procedure
input_name;
begin
    name:='';
    cleardevice;
    setbkcolor(green);
    setttextstyle(7,0,6);
    setcolor(white);
    outtextxy(10,100,'Введите Ваше имя');
    rectangle(10,170,629,250);
    outtextxy(220,300,'Далее');
    setfillstyle(1,white);
    bar(190,380,440,450);
    setttextstyle(0,0,6);
    setcolor(brown);
    outtextxy(200,390,'ENTER');

```

```

settextstyle(0,0,6);
setcolor(yellow);
setfillstyle(green,0);
repeat
    key:=readkey;
    if (key=backspace)
    then
        begin
            if (ord(name[0])>0)
            then
                begin
                    name[0]:=chr(ord(name[0]) - 1);
                    bar(11,171,628,249);
                    outtextxy(30,190,name)
                end
            end
        else
            begin
                if ((ord(name[0])<12) and (key<>enter))
                then
                    begin
                        name:=name+key;
                        outtextxy(30,190,name);
                    end
                end;
            if ((ord(name[0])=0) and (key=enter))
            then
                begin
                    setfillstyle(1,8);
                    bar(50,150,600,400);
                    settextstyle(1,0,7);
                    setcolor(white);
                    outtextxy(190,175,'Ошибка');
                    settextstyle(1,0,6);
                    outtextxy(55,250,'Повторите попытку');
                    key:=readkey;
                    input_name;
                end;
            until (key=enter);
        exit;
    end;
end;

```

```

procedure
top;
type
    tops=record
        name,temp:string;
        scores:integer;
    end;
    mas=array[1..20] of tops;
var
    f:text;
    a:mas;
    s,temp,temp_name:string;
    i:byte;
    code,t,j:integer;
begin
    j:=1;
    for i:=1 to 20 do

```

```

begin
    a[i].name:='';
    a[i].scores:=0;
end;
assign(f, 'y:/kolobok/results.txt');
{$i-};
reset(f);
{$i+};
if ioresult<>0
then
    begin
        setfillstyle(1,8);
        bar(50,150,600,400);
        setcolor(white);
        settextstyle(0,0,2);
        outtextxy(70,175,'Произошла ошибка');
        outtextxy(70,200,'Отсутствует файл с результатами');
        setcolor(yellow);
        settextstyle(0,0,6);
        outtextxy(270,300,'OK');
    end
else
    begin
        while not eof(f) do
            begin
                temp:='';
                temp_name:='';
                readln(f,s);
                for i:=7 to length(s) do
                    temp_name:=temp_name+s[i];
                for i:=2 to 5 do
                    readln(f,s);
                readln(f,s);
                for i:=8 to length(s) do
                    temp:=temp+s[i];
                val(temp,t,code);
                readln(f,s);
                a[j].name:=temp_name;
                a[j].scores:=t;
                j:=j+1;
            end;
        close(f);
        for i:=1 to 19 do
            for j:=i+1 to 20 do
                if a[i].scores<a[j].scores
                then
                    begin
                        t:=a[i].scores;
                        a[i].scores:=a[j].scores;
                        a[j].scores:=t;
                        temp_name:=a[i].name;
                        a[i].name:=a[j].name;
                        a[j].name:=temp_name;
                    end;
            cleardevice;
            setbkcolor(green);
            setcolor(yellow);
            settextstyle(0,0,7);
            outtextxy(180,10,'ТОП-5');
            settextstyle(0,0,5);

```



```

        outtextxy(280,350,'OK');
        setcolor(white);
        line(0,70,650,70);
        line(0,340,650,340);
        line(320,70,320,340);
        settextstyle(7,0,4);
        outtextxy(100,75,'Имя');
        outtextxy(380,75,'Результат');
        line(0,115,640,115);
        settextstyle(0,0,4);
        j:=120;
        for i:=1 to 5 do
            if (length(a[i].name)>0)
            then
                begin
                    str(a[i].scores,temp);
                    outtextxy(20,j,a[i].name);
                    outtextxy(450,j,temp);
                    j:=j+40;
                end;
            setfillstyle(1,white);
            bar(200,400,450,470);
            settextstyle(0,0,6);
            setcolor(brown);
            outtextxy(210,415,'ENTER');
        end;
    end;
end;

```

```

procedure
menu;
var
    c:byte;
begin
    cleardevice;
    setbkcolor(green);
    setcolor(yellow);
    settextstyle(0,0,10);
    outtextxy(50,50,'КОЛОВОК');
    setcolor(yellow);
    settextstyle(7,0,6);
    outtextxy(200,140,'Играть');
    c:=1;
    setcolor(white);
    outtextxy(220,300,'ТОП-5');
    outtextxy(130,220,'Инструкции');
    outtextxy(210,370,'Выход');
    repeat
        key:=readkey;
        if (((c=1) and (key=down)) or ((c=3) and (key=up)))
        then
            begin
                setcolor(yellow);
                outtextxy(130,220,'Инструкции');
                setcolor(white);
                outtextxy(200,140,'Играть');
                outtextxy(210,370,'Выход');
                outtextxy(220,300,'ТОП-5');
                c:=2;
            end
        end
    end;
end;

```

```

else
  if ((c=3) and (key=down))
  then
    begin
      setcolor(yellow);
      outtextxy(210,370,'Выход');
      setcolor(white);
      outtextxy(130,220,'Инструкции');
      outtextxy(200,140,'Играть');
      outtextxy(220,300,'ТОП-5');
      c:=4;
    end
  else
    if ((c=2) and (key=up))
    then
      begin
        setcolor(yellow);
        outtextxy(200,140,'Играть');
        setcolor(white);
        outtextxy(130,220,'Инструкции');
        outtextxy(210,370,'Выход');
        outtextxy(220,300,'ТОП-5');
        c:=1;
      end
    else
      if (((c=2) and (key=down)) or ((c=4) and (key=up)))
      then
        begin
          setcolor(yellow);
          outtextxy(220,300,'ТОП-5');
          outtextxy(200,140,'Играть');
          setcolor(white);
          outtextxy(200,140,'Играть');
          outtextxy(130,220,'Инструкции');
          outtextxy(210,370,'Выход');
          c:=3;
        end;
      until (key=enter);
      if (c=4)
      then
        begin
          repeat
            exit;
            key:=readkey;
            if ((key<>y1) and (key<>y2) and (key<>n1) and (key<>n2)
and (key<>y1) and (key<>y2))
            then
              begin
                sound(400);
                delay(150);
                nosound;
              end
            else
              if ((key=n1) or (key=n2))
              then
                begin
                  cleardevice;
                  menu;
                  c:=1;
                  exit;
                end
              end
            end
          until (key=enter);
        end
      end
    end
  end
end

```

```

        end
    else
        begin
            closegraph;
            exit;
        end
    until ((key=n1) or (key=n2) or (key=y1) or (key=y2))
end
else
    if c=2
    then
        begin
            repeat
                instructions;
                key:=readkey;
            until (key=enter);
            menu
        end
    else
        if (c=1)
        then
            begin
                input_name;
                exit;
            end
        else
            if c=3
            then
                begin
                    repeat
                        top;
                        key:=readkey;
                    until (key=enter);
                    menu;
                end;
            end;
        end;
    end;

begin
    clrscr;
    gd:=detect;
    initgraph(gd,gm,'');
    if errorcode=0
    then
        begin
            cleardevice;
            greeting;
            repeat
                key:=readkey;
                if ( (key<>enter))
                then
                    begin
                        repeat
                            exitt;
                            key:=readkey;
                            if ((key<>y1) and (key<>y2) and (key<>n1) and
(key<>n2) and (key<>y1) and (key<>y2))
                            then
                                sound(400);
                                delay(150);

```

```

        nosound;
    if ((key<>y1) and (key<>y2))
    then
        begin
            cleardevice;
            greeting
        end
    else
        closegraph;
    until ((key=n1) or (key=n2) or (key=y1) or (key=y2))
end
else
begin
    repeat
        menu;
        input_positive_count;
        input_negative_count;
        input_mas;
        game;
        output_data;
    until (c<>0);
end
until (key=enter);
end
else
begin
    writeln('ERROR');
    writeln(grapherrormsg(errorcode));
    halt(1);
end;
closegraph;
end.

```

## Приложение 2 Скриншоты программы

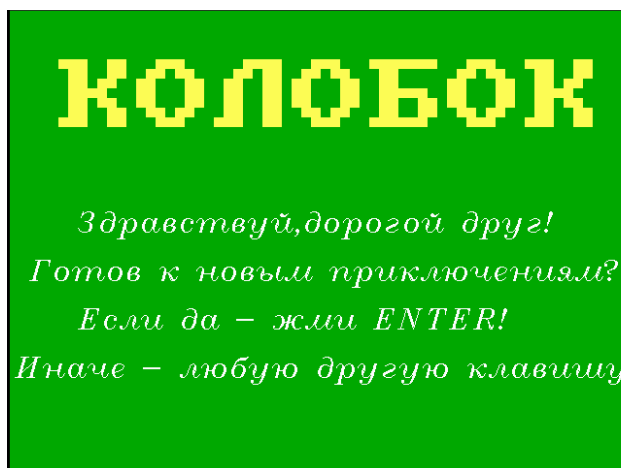


Рисунок 23 Приветствие

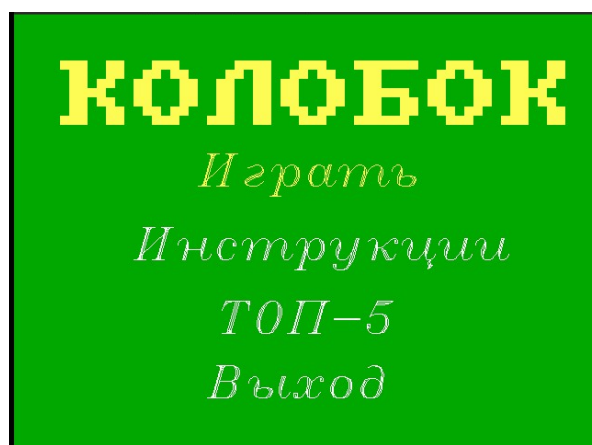


Рисунок 24 Меню

ТОП-5	
Имя	Результат
xxx	7
fghjkl;	5
juletz	4
hi	3
OK	
ENTER	

Рисунок 25 Топ-5 игроков

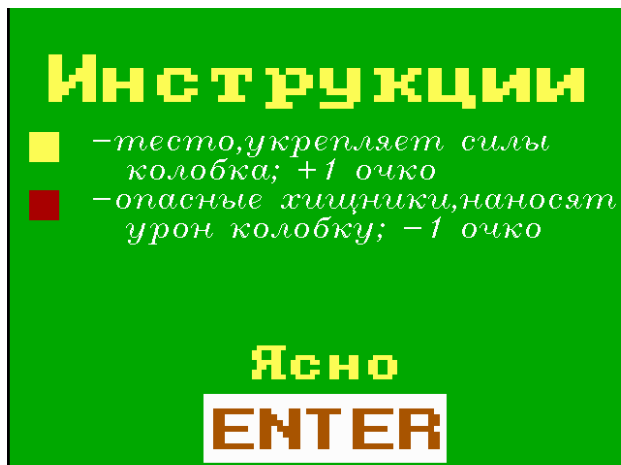


Рисунок 26 Инструкции



Рисунок 27 Ввод имени

Количество отрицательных препятствий	Количество положительных препятствий
5	5
10	10
15	15

Рисунок 28 Ввод количества препятствий

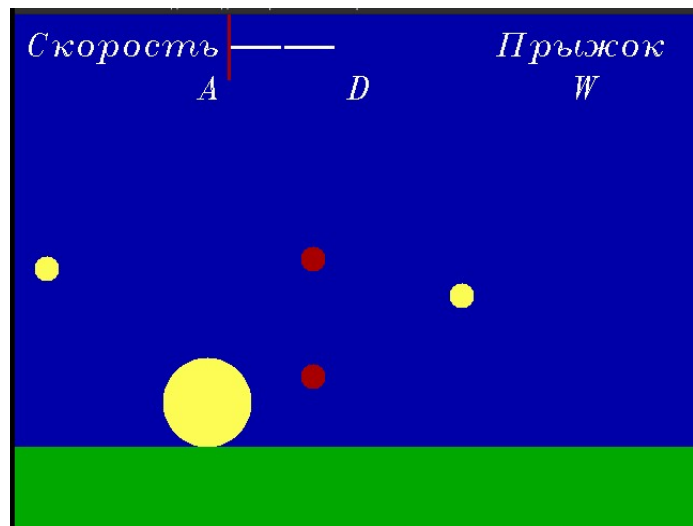


Рисунок 29 Игровой процесс

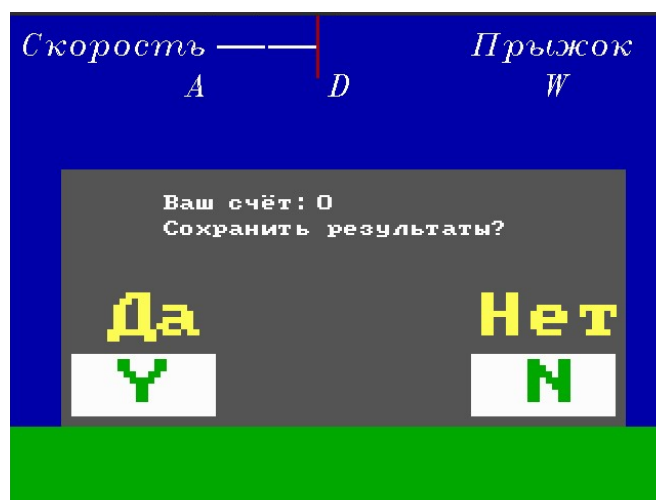


Рисунок 30 Сохранение результатов

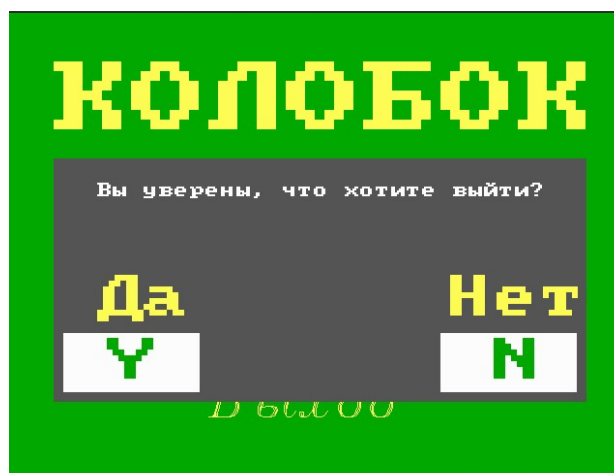


Рисунок 31 Выход из игры