**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА**

**(РОСАВИАЦИЯ)**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)

Кафедра вычислительных машин, комплексов, сетей и систем.

Курсовая работа

защищена с оценкой

(подпись преподавателя, дата)

Курсовая работа по курсу

«Операционные системы»

«Создание Win32 приложения с использованием среды “Python”.»

Выполнила:

студентка группы ЭВМ 1-1

Торяник Валерия Евгеньевна

Руководитель:

к. ф. м. н. доцент

Черкасова Наталья Ивановна

МОСКВА - 2021

**Содержание:**

[Аннотация: 3](#_Toc71197609)

[Техническое задание: 4](#_Toc71197610)

[1.Краткие теоретические сведения: 5](#_Toc71197611)

[2. Разработка приложения: 8](#_Toc71197612)

[3. Алгоритм работы программы 9](#_Toc71197613)

[4. Системные требования 17](#_Toc71197614)

[5. Руководство пользователя 17](#_Toc71197615)

[6. Список используемой литературы. 20](#_Toc71197616)

[7. Приложение. Листинг программы. 20](#_Toc71197617)

## Аннотация:

В рамках курсовой работы было разработано приложение-игра «Крестики-нолики» с использованием среды Python.

Игрок и компьютер по очереди ставят на свободные клетки поля 3х3 знаки (один всегда крестики, другой всегда нолики). Первый, выстроивший в ряд 3 своих фигуры по вертикали, горизонтали или диагонали, выигрывает. Первый ход делает игрок, ставящий крестики.

Помимо игры можно производить следующие действия: сведения о программе, правила игры, отчистку игрового поля и начало новой игры, выход из приложения.

Отчёт по результатам работы состоит из графической части пояснительной записки.

Пояснительная записка включает в себя техническое задание, структуру программы, информацию о системных требованиях и руководство пользователя. Графическая часть включает в себя листинги программы и результаты выполнения программы.

## Техническое задание:

Разработать приложение-игру «Крестики-нолики» с использованием среды Python.

Игрок и компьютер по очереди ставят на свободные клетки поля 3х3 знаки (один всегда крестики, другой всегда нолики). Первый, выстроивший в ряд 3 своих фигуры по вертикали, горизонтали или диагонали, выигрывает. Первый ход делает игрок, ставящий крестики.

Помимо игры можно производить следующие действия: сведения о программе, правила игры, отчистку игрового поля и начало новой игры, выход из приложения.

## Краткие теоретические сведения:

**1.1.** **Особенности создания Win-приложений.**

Windows - приложения выполняются в собственных окнах. Каждое приложение располагает, по крайней мере, одним собственным окном. Через окна приложения выполняется ввод/вывод информации пользователя. Главное окно – это и есть само приложение, но окно – это также и визуальный интерфейс.

Работа в Windows ориентирована на события. В Windows приложения выполняются пошагово. После решение одной подзадачи, управление возвращается Windows, которая может вызывать другие программы. Windows переключается между различными приложениями. Программист инициирует событие (вызов команды меню, щелчок мыши на окне), событие обрабатывается, и программное управление передается в соответствующее приложение. Приложение вызывается для обработки события.

Таким образом, разработка приложения – это создание окна приложения (создать окно, зарегистрировать его класс, сделать его видимым) и организация обработки сообщений пользователя.

Сообщения Windows – это способ передачи информации приложению. Каждое сообщение отражает некоторое системное событие, на которое приложение может отреагировать. Каждое сообщение представляется в программе структурой, которая содержит адрес окна (дескриптор), которое должно принять это сообщение, имя сообщения и некоторые параметры для передачи окну информации.

Операционная система Windows посылает каждому приложению сообщения обо всех относящихся к нему системных событиях. К этим событиям относятся, например, перемещение пользователем мыши, нажатие клавиши на клавиатуре, изменение размеров и перемещение окна, манипуляция с элементами управления в окне приложения.

Сгенерировав (заполнив структуры) эти сообщения, Windows помещает их в ядро системы, в системную очередь сообщений. Ядро извлекает каждое сообщение и пересылает его в очередь сообщений нужного приложения, в соответствии с дескриптором окна, содержащемся в каждом сообщении. Т.е. по дескриптору определяется нужное приложение.

**1.2. Язык программирования Python.**

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Язык является полностью объектно-ориентированным — всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации, сам же язык известен как интерпретируемый и используется в том числе для написания скриптов.

Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как Си или C++.

Python является мультипарадигмальным языком программирования, поддерживающим императивное, процедурное, структурное, объектно-ориентированное программирование, метапрограммирование и функциональное программирование.

Стандартная библиотека включает большой набор полезных переносимых функций, начиная от функционала для работы с текстом и заканчивая средствами для написания сетевых приложений. Дополнительные возможности, такие как математическое моделирование, работа с оборудованием, написание веб-приложений или разработка игр, могут реализовываться посредством обширного количества сторонних библиотек, а также интеграцией библиотек, написанных на Си или C++, при этом и сам интерпретатор Python может интегрироваться в проекты, написанные на этих языках.

За счёт читабельности, простого синтаксиса и отсутствия необходимости в компиляции язык хорошо подходит для обучения программированию, позволяя концентрироваться на изучении алгоритмов, концептов и парадигм. Отладка же и экспериментирование в значительной степени облегчаются тем фактом, что язык является интерпретируемым (интерпретатор CPython)

**1.3. Среда разработки PyCharm.**



PyCharm - это интегрированная среда разработки для Python, которая имеет полный комплект средств, необходимых для эффективного программирования на Python.

PyCharm предоставляет умную проверку кода, быстрое выявление ошибок и оперативное исправление, вкупе с автоматическим рефакторингом кода, и богатыми возможностями в навигации. Кроме этого, скачать PyCharm не составит никакого труда.

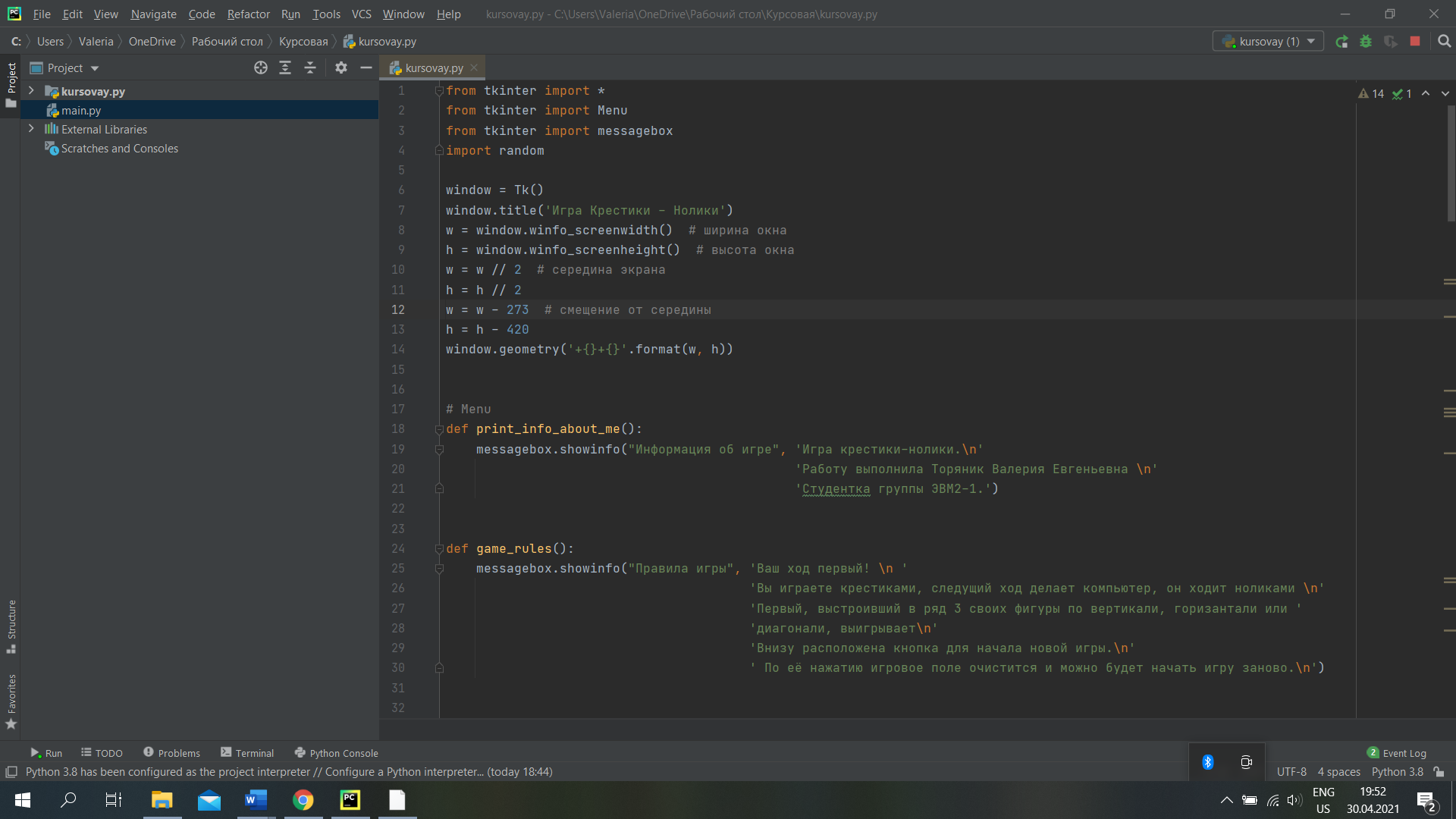


Рисунок 1.

Возможности PyCharm:

* Мощный и функциональный редактор кода с подсветкой синтаксиса, авто-форматированием и авто-отступами для поддерживаемых языков; простая и мощная навигация в коде.
* Помощь при написании кода, включающая в себя автодополнение, авто-импорт, шаблоны кода, проверка на совместимость версии интерпретатора языка, и многое другое.
* Быстрый просмотр документации для любого элемента прямо в окне редактора, просмотр внешней документации через браузер, поддержка docstring – генерация, подсветка, автодополнение и многое другое.
* Большое количество инспекций кода.
* Мощный рефакторинг кода, который предоставляет широкие возможности по выполнению быстрых глобальных изменений в проекте.

В курсовой работе используются дополнительные библиотеки:

* Tkinter - стандартная библиотека для отображения графики.
* random - получение случайных чисел, благодаря которым ходы компьютера будут неожиданными.

## 2. Разработка приложения:

**2.1. Пользовательские функции приложения:**

В таблице 1 представлены пользовательские функции приложения:

|  |  |
| --- | --- |
| ТТаблица 1: |  |
| Название: | Назначение: |
| print\_info\_about\_me | Вывод информации о программе в диалоговом окне. |
| game\_rules | Вывод правил игры в диалоговом окне. |
| new\_game | Очистка игрового поля и начало новой игры. |
| click | Ход игрока. |
| check\_win | Проверка выигрыша. |
| check\_line | Перебор всех возможных комбинаций поля для  проверки выигрыша (находится внутри ф. check\_win) |
| can\_win | Препятствие со стороны компьютера выиграть в один ход игроку. |
| computer\_move | Выполнение выигрышного или случайного хода компьютера. |

## 

## 3. Алгоритм работы программы

**Алгоритм № 1.**

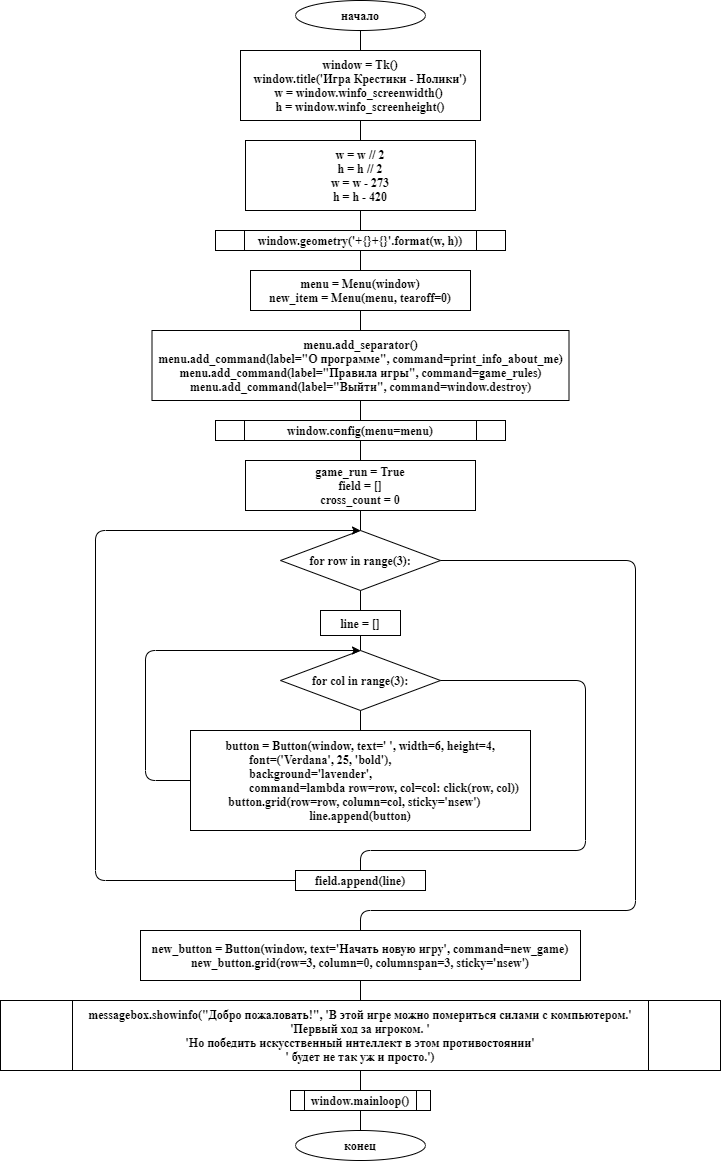


Рисунок 2.

**Алгоритм № 2.**

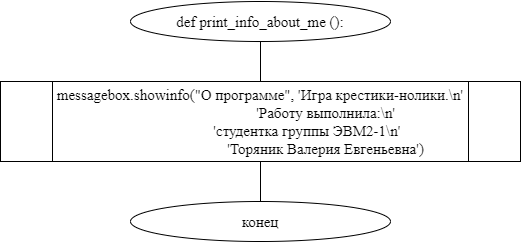


Рисунок 3.

**Алгоритм № 3.**

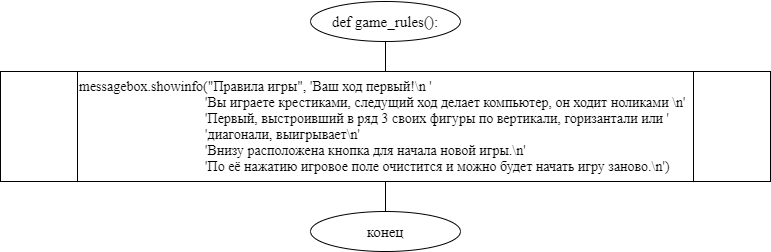


Рисунок 4.

**Алгоритм № 4.**

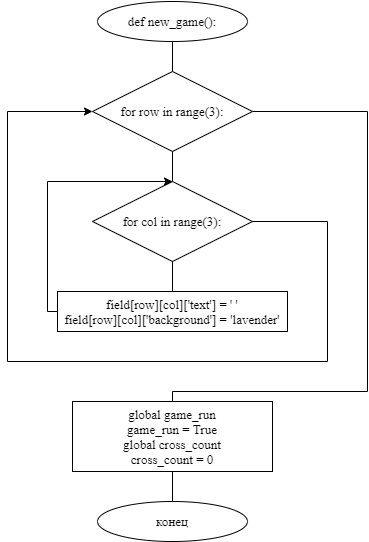


Рисунок 5.

**Алгоритм № 5.**

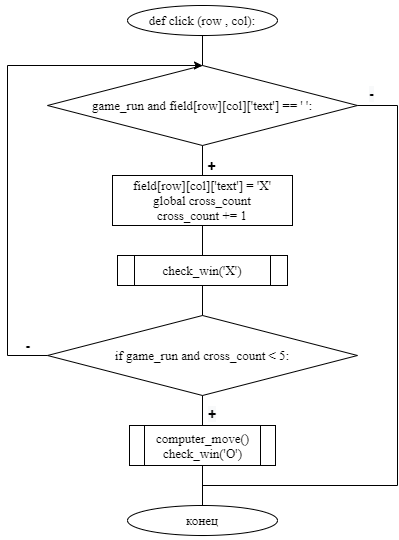


Рисунок 6.

**Алгоритм № 6.**

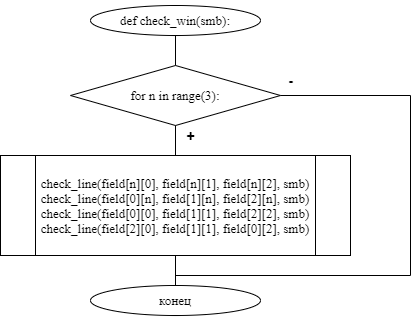


Рисунок 7.

**Алгоритм № 7.**

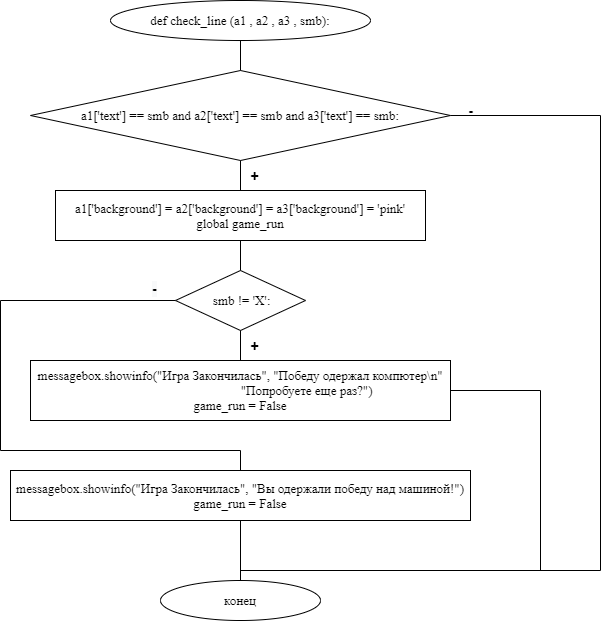


Рисунок 8.

**Алгоритм № 8.**

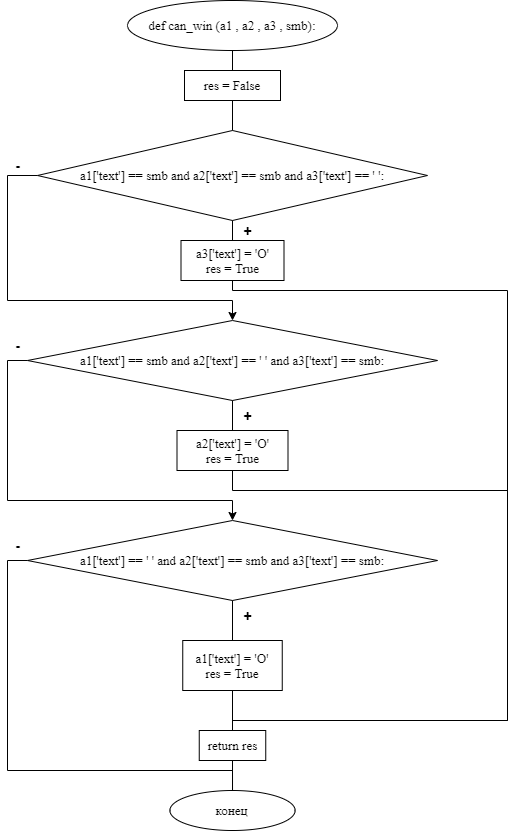
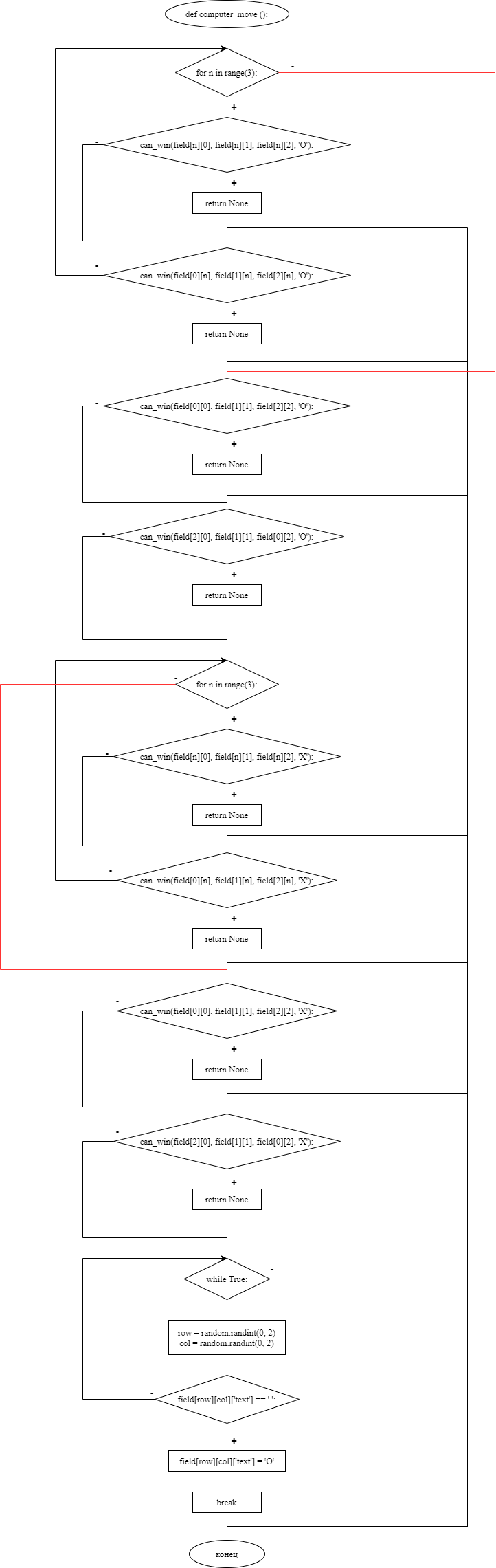


Рисунок 9.

**Алгоритм № 9.**

Рисунок 10.

## 4. Системные требования

Для работы программы необходима следующая минимальная конфигурация системы:

* Windows 7 и выше;
* Не менее 9,5 Мб свободного пространства на диске;
* Библиотеки Tkinter и random.

## 5. Руководство пользователя

Запустить файл kursovay.exe.

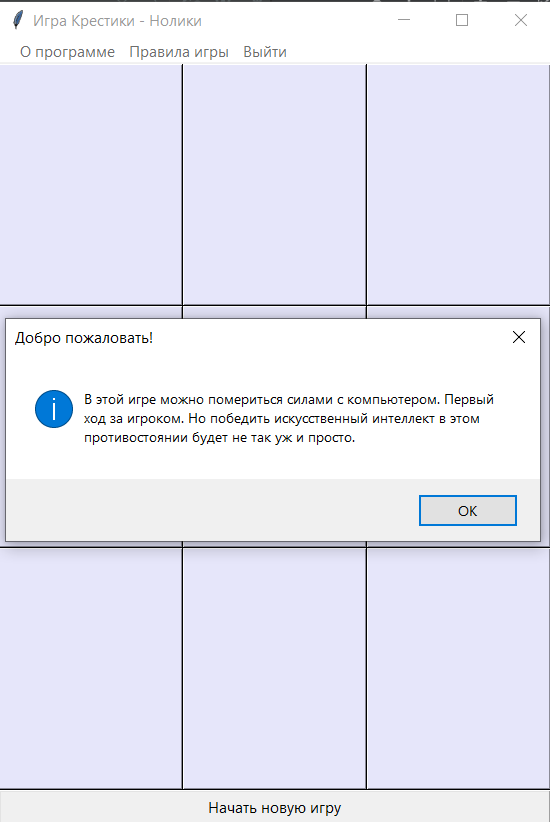


Рисунок 11.

Нажимая кнопку OK - игра начинается. Игрок ставит крестики, а компьютер нолики. Побеждает тот, кто первый составит линию из своих символов: по горизонтали, вертикали или диагонали.

Вверху расположены кнопки «О программе», «Правила игры» и «Выход», нажимая каждую из них можно узнать кто разработал данную игру (рисунок 12), как в нее правильно играть (рисунок 13) и совершить выход из игры.

Внизу расположена кнопка «Начать новую игру». По её нажатию игровое поле очистится и можно будет начать игру заново.

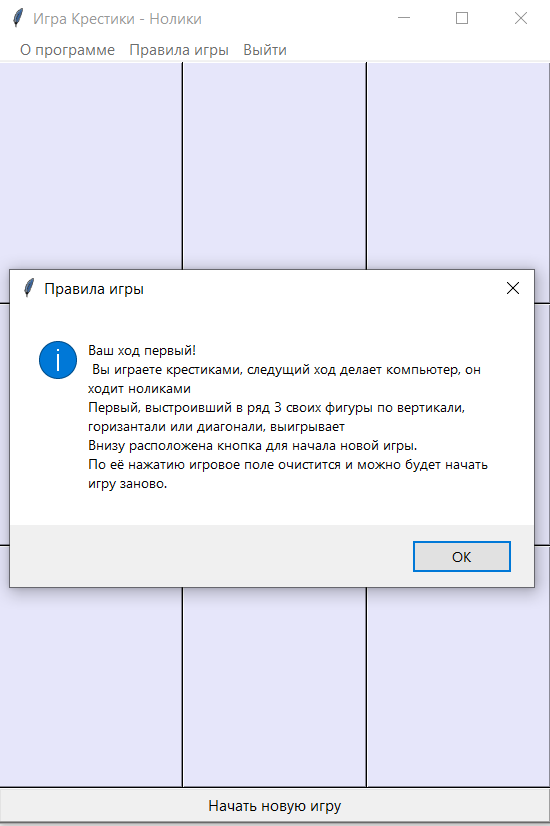
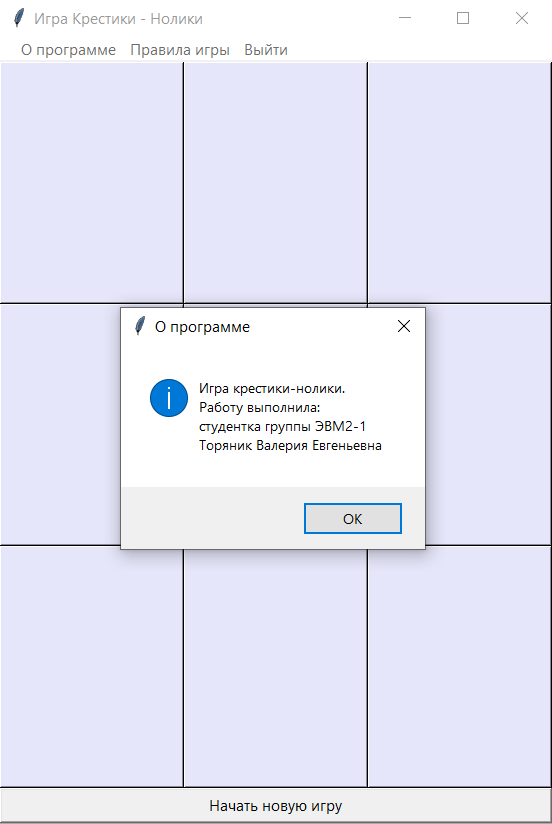


Рисунок 12. Рисунок 13.

При победе компьютера или игрока высвечивается диалоговое окно с информацией о выигрыше (рисунок 14,15)

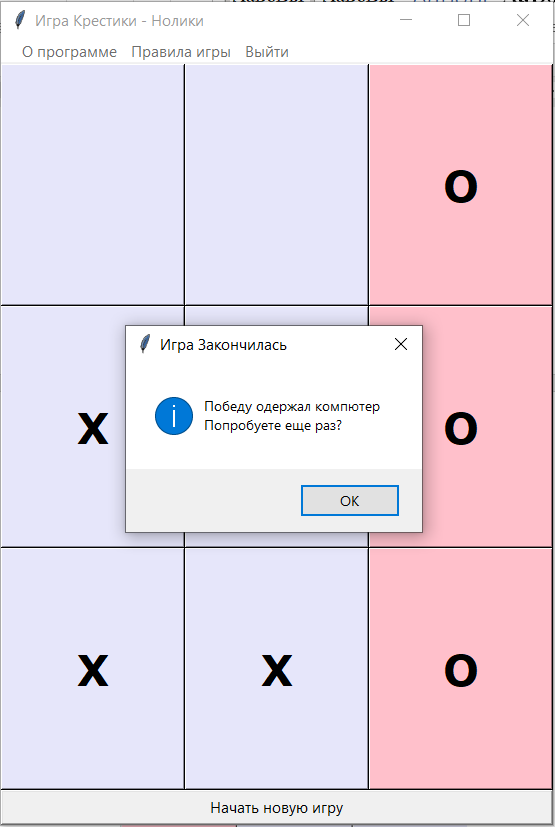
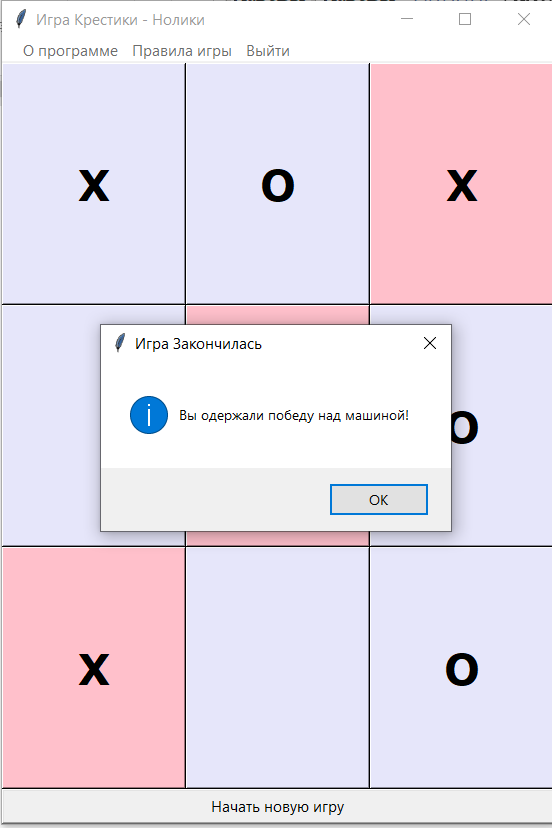


Рисунок 14. Рисунок 15.

## 6. Список используемой литературы.

1. Черкасова Н. И., Надейкина Л.А. Операционные системы: пособие по выполнению курсовой работы. – М.: МГТУ ГА, 2014. - 28с.

2. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. - СПб.: БХВ-Петербург, 2014. - 704 с.

3. Python [Электронный ресурс] - Электрон. дан. - Режим доступа: <https://www.python.org>

## 7. Приложение. Листинг программы.

from tkinter import \*

from tkinter import Menu

from tkinter import messagebox

import random

window = Tk()

window.title('Игра Крестики - Нолики')

w = window.winfo\_screenwidth() # ширина окна

h = window.winfo\_screenheight() # высота окна

w = w // 2 # середина экрана

h = h // 2

w = w - 273 # смещение от середины

h = h - 420

window.geometry('+{}+{}'.format(w, h))

# Menu

def print\_info\_about\_me():

messagebox.showinfo("О программе", 'Игра крестики-нолики.\n'

'Работу выполнила:\n'

'студентка группы ЭВМ2-1\n'

'Торяник Валерия Евгеньевна')

def game\_rules():

messagebox.showinfo("Правила игры", 'Ваш ход первый!\n '

'Вы играете крестиками, следущий ход делает компьютер, он ходит ноликами \n'

'Первый, выстроивший в ряд 3 своих фигуры по вертикали, горизантали или '

'диагонали, выигрывает\n'

'Внизу расположена кнопка для начала новой игры.\n'

'По её нажатию игровое поле очистится и можно будет начать игру заново.\n')

menu = Menu(window)

new\_item = Menu(menu, tearoff=0)

menu.add\_separator()

menu.add\_command(label="О программе", command=print\_info\_about\_me)

menu.add\_command(label="Правила игры", command=game\_rules)

menu.add\_command(label="Выйти", command=window.destroy)

window.config(menu=menu)

game\_run = True

field = []

cross\_count = 0

def new\_game():

for row in range(3):

for col in range(3):

field[row][col]['text'] = ' '

field[row][col]['background'] = 'lavender'

global game\_run

game\_run = True

global cross\_count

cross\_count = 0

def click(row, col):

if game\_run and field[row][col]['text'] == ' ':

field[row][col]['text'] = 'X'

global cross\_count

cross\_count += 1

check\_win('X')

if game\_run and cross\_count < 5:

computer\_move()

check\_win('O')

def check\_win(smb):

for n in range(3):

check\_line(field[n][0], field[n][1], field[n][2], smb)

check\_line(field[0][n], field[1][n], field[2][n], smb)

check\_line(field[0][0], field[1][1], field[2][2], smb)

check\_line(field[2][0], field[1][1], field[0][2], smb)

def check\_line(a1, a2, a3, smb):

if a1['text'] == smb and a2['text'] == smb and a3['text'] == smb:

a1['background'] = a2['background'] = a3['background'] = 'pink'

global game\_run

if smb != 'X':

messagebox.showinfo("Игра Закончилась", "Победу одержал компютер\n"

"Попробуете еще раз?")

game\_run = False

else:

messagebox.showinfo("Игра Закончилась", "Вы одержали победу над машиной!")

game\_run = False

def can\_win(a1, a2, a3, smb):

res = False

if a1['text'] == smb and a2['text'] == smb and a3['text'] == ' ':

a3['text'] = 'O'

res = True

if a1['text'] == smb and a2['text'] == ' ' and a3['text'] == smb:

a2['text'] = 'O'

res = True

if a1['text'] == ' ' and a2['text'] == smb and a3['text'] == smb:

a1['text'] = 'O'

res = True

return res

def computer\_move():

for n in range(3):

if can\_win(field[n][0], field[n][1], field[n][2], 'O'):

return

if can\_win(field[0][n], field[1][n], field[2][n], 'O'):

return

if can\_win(field[0][0], field[1][1], field[2][2], 'O'):

return

if can\_win(field[2][0], field[1][1], field[0][2], 'O'):

return

for n in range(3):

if can\_win(field[n][0], field[n][1], field[n][2], 'X'):

return

if can\_win(field[0][n], field[1][n], field[2][n], 'X'):

return

if can\_win(field[0][0], field[1][1], field[2][2], 'X'):

return

if can\_win(field[2][0], field[1][1], field[0][2], 'X'):

return

while True:

row = random.randint(0, 2)

col = random.randint(0, 2)

if field[row][col]['text'] == ' ':

field[row][col]['text'] = 'O'

break

for row in range(3):

line = []

for col in range(3):

button = Button(window, text=' ', width=6, height=4,

font=('Verdana', 25, 'bold'),

background='lavender',

command=lambda row=row, col=col: click(row, col))

button.grid(row=row, column=col, sticky='nsew')

line.append(button)

field.append(line)

new\_button = Button(window, text='Начать новую игру', command=new\_game)

new\_button.grid(row=3, column=0, columnspan=3, sticky='nsew')

messagebox.showinfo("Добро пожаловать!", 'В этой игре можно помериться силами с компьютером. Первый ход за игроком. '

'Но победить искусственный интеллект в этом противостоянии будет не так уж и '

'просто.')

window.mainloop()