МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

(ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение»

Направление подготовки 09.03.04 «Программная инженерия»

ОТЧЕТ ПО ПРАКТИКЕ

|  |
| --- |
|  |
| *(наименование типа практики)* |
| *(полное наименование профильной организации)*  Выполнил обучающийся /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / |
| *(подпись) (ФИО, курс, номер группы)*  Дата сдачи отчета: « » 202 г. |
| Дата аттестации « » 202 г. |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Руководитель практики от  ИЖГТУ имени М.Т. Калашникова / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ |
| *(подпись) (И.О. Фамилия, должность, ученая степень)*  Заведующий кафедрой /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ |
| *(подпись) (И.О. Фамилия, должность, ученая степень)* |

СОдержание

[Введение 3](#_Toc168145474)

[1. Основная часть 4](#_Toc168145475)

[1.1 Словесное описание проекта 4](#_Toc168145476)

[1.2 Математическая постановка 4](#_Toc168145477)

[1.3 Описание программной реализации 5](#_Toc168145478)

[1.4 Листинг текста программы, тестовые примеры работы 5](#_Toc168145479)

[Заключение 10](#_Toc168145480)

[Список использованных источников 11](#_Toc168145481)

# Введение

В данной работе представлена разработка компьютерной версии классической игры "Крестики-Нолики", выполненная с использованием языка программирования Python и библиотеки Tkinter для создания графического пользовательского интерфейса. Целью проекта является создание полноценной и интерактивной игры, в которой два пользователя поочередно ставят крестики и нолики на игровом поле размером 3x3. Проект охватывает процессы проектирования интерфейса, реализации игровой логики и обеспечения корректной работы игры.

# 1. Основная часть

1.1 Словесное описание проекта

Главная цель проекта – создать компьютерную версию популярной игры "Крестики-Нолики", используя язык программирования Python и библиотеку Tkinter для создания графического интерфейса. Программа должна:

Создавать игровое поле размером 3x3.

Реализовывать логику игры, включая ввод хода игрока, проверку на победу и ничью.

Обеспечить пользователю возможность перезапуска игры после завершения текущей партии.

Создать графическое приложение на языке программирования Python.

Реализовать игровое поле размером 3x3 с использованием кнопок Tkinter.

Организовать ввод ходов двух игроков, чередуя их.

Проверять состояние игрового поля после каждого хода для определения победы или ничьей.

Отображать игровое поле после каждого хода.

Предоставить возможность перезапуска игры после завершения партии.

1.2 Математическая постановка

Основная игровая логика заключается в следующем:

Игровое поле представлено матрицей размером 3x3.

Два игрока поочередно ставят свои метки (крестики 'X' и нолики 'O') на свободные клетки игрового поля.

Проверка на победу осуществляется после каждого хода. Победа достигается, если три одинаковые метки располагаются в ряд по горизонтали, вертикали или диагонали.

Если все клетки заполнены и ни один из игроков не выиграл, объявляется ничья.

1.3 Описание программной реализации

Программа представляет собой объектно-ориентированное решение:

Класс TicTacToe:

Атрибуты:

self.root: главное окно приложения.

self.player: текущий игрок ('X' или 'O').

self.buttons: матрица кнопок игрового поля размером 3x3.

Методы:

\_\_init\_\_: инициализация игры, создание кнопок и настройка сетки.

create\_buttons: создание кнопок игрового поля.

configure\_grid: настройка строк и столбцов для растяжения при изменении размера окна.

on\_button\_click: обработка нажатий на кнопки, проверка на победу или ничью, смена игрока.

check\_winner: проверка победителя по горизонтали, вертикали и диагонали.

check\_draw: проверка ничьи.

reset\_board: сброс игрового поля для новой игры.

1.4 Листинг текста программы, тестовые примеры работы

import tkinter as tk  
from tkinter import messagebox  
  
class TicTacToe:  
 def \_\_init\_\_(self, root):  
 self.root = root  
 self.root.title("Крестики-Нолики")  
  
 self.player = "X" # Начальный игрок  
 self.buttons = [[None for \_ in range(3)] for \_ in range(3)] # Сетка кнопок 3x3  
  
 self.create\_buttons()  
 self.configure\_grid()  
  
 def create\_buttons(self):  
 # Создание кнопок и размещение их на сетке  
 for i in range(3):  
 for j in range(3):  
 button = tk.Button(self.root, text="", font=('normal', 40),  
 command=lambda i=i, j=j: self.on\_button\_click(i, j))  
 button.grid(row=i, column=j, sticky="nsew")  
 self.buttons[i][j] = button  
  
 def configure\_grid(self):  
 # Настройка строк и столбцов сетки для растяжения при изменении размера окна  
 for i in range(3):  
 self.root.grid\_rowconfigure(i, weight=1)  
 self.root.grid\_columnconfigure(i, weight=1)  
  
 def on\_button\_click(self, i, j):  
 # Обработка нажатия на кнопку  
 if self.buttons[i][j]['text'] == "" and not self.check\_winner():  
 self.buttons[i][j]['text'] = self.player  
 if self.check\_winner():  
 messagebox.showinfo("Победа!", f"Игрок {self.player} победил!")  
 self.reset\_board()  
 elif self.check\_draw():  
 messagebox.showinfo("Ничья", "Ничья!")  
 self.reset\_board()  
 else:  
 # Смена игрока  
 self.player = "O" if self.player == "X" else "X"  
  
 def check\_winner(self):  
 # Проверка победителя с использованием рекурсии  
 for i in range(3):  
 if self.check\_winner\_recursive(i, 0, 0, 1) or self.check\_winner\_recursive(0, i, 1, 0):  
 return True  
 if self.check\_winner\_recursive(0, 0, 1, 1) or self.check\_winner\_recursive(0, 2, 1, -1):  
 return True  
 return False  
  
 def check\_winner\_recursive(self, row, col, row\_delta, col\_delta, count=0):  
 # Рекурсивная проверка победителя  
 if row < 0 or row >= 3 or col < 0 or col >= 3:  
 return False  
 if self.buttons[row][col]['text'] == self.player:  
 count += 1  
 if count == 3:  
 return True  
 return self.check\_winner\_recursive(row + row\_delta, col + col\_delta, row\_delta, col\_delta, count)  
 return False  
  
 def check\_draw(self):  
 # Проверка, есть ли ничья  
 for row in self.buttons:  
 for button in row:  
 if button['text'] == "":  
 return False  
 return True  
  
 def reset\_board(self):  
 # Сброс игрового поля для новой игры  
 for row in self.buttons:  
 for button in row:  
 button['text'] = ""  
 self.player = "X" # Возврат начального игрока  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 root = tk.Tk() # Создание основного окна приложения  
 game = TicTacToe(root) # Создание экземпляра игры  
 root.mainloop() # Запуск основного цикла приложения

Тестовые примеры работы:



Рис. 1 (начальное окно)

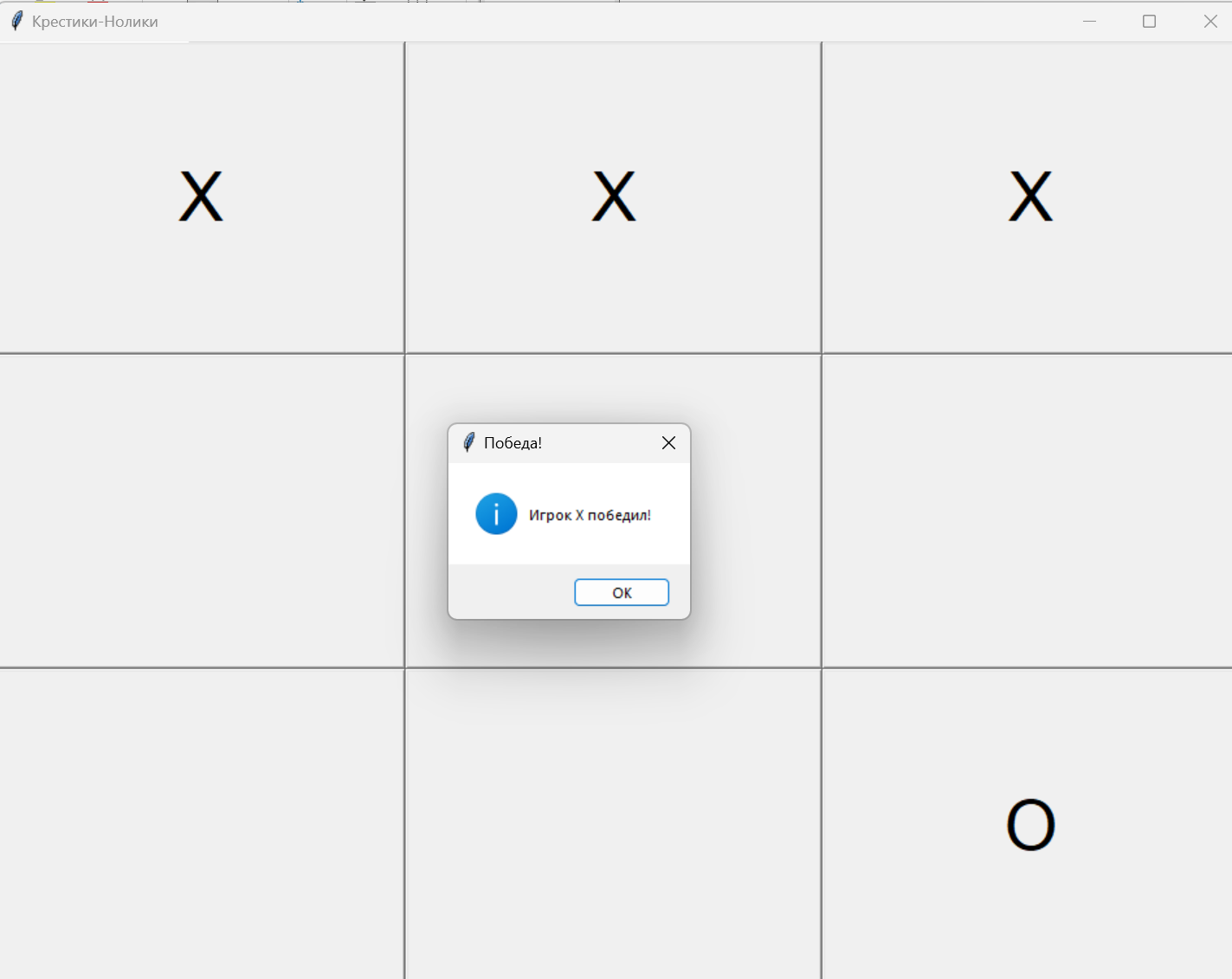


Рис. 2 (Победа игрока Х)

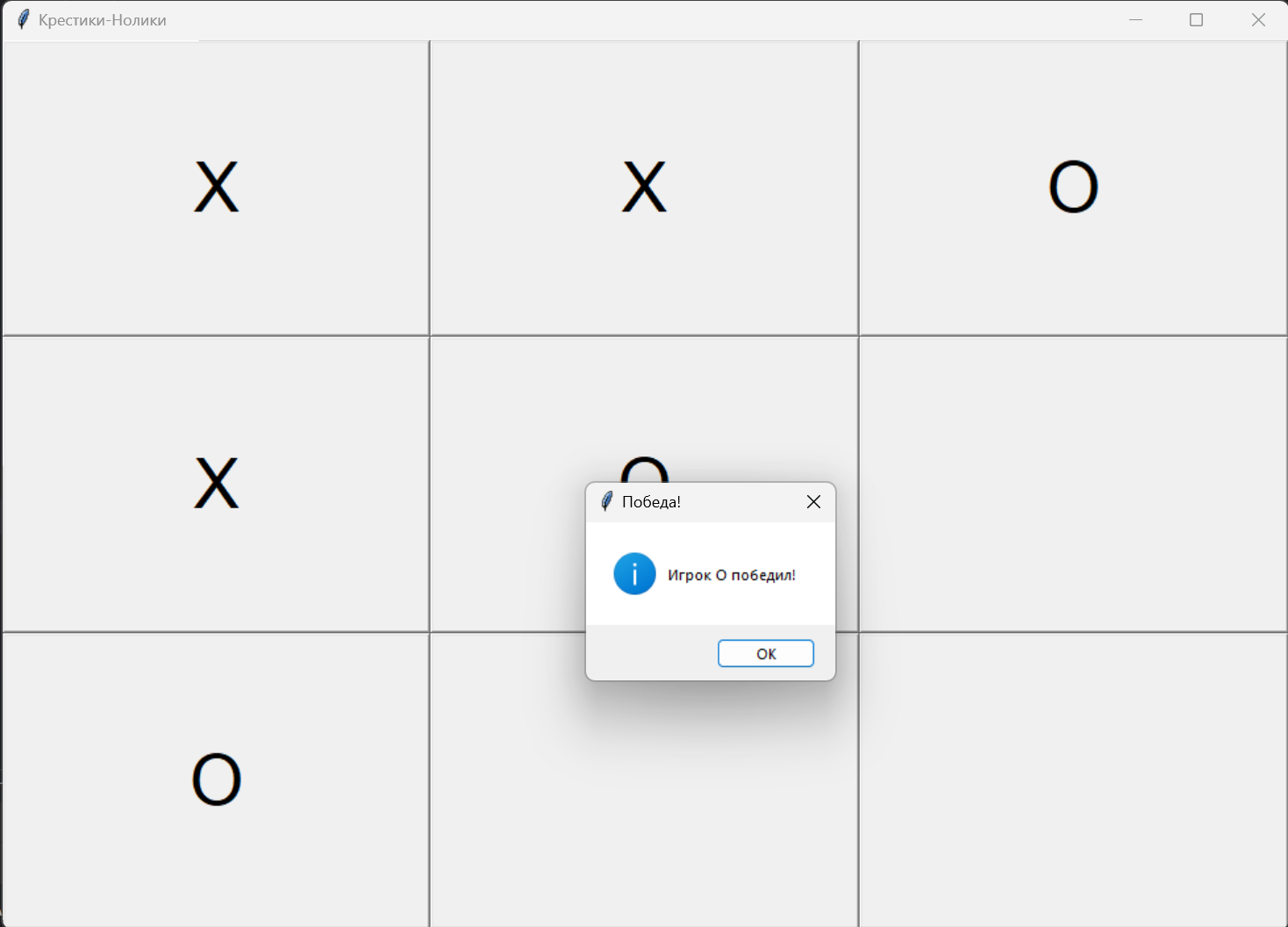


Рис. 3 (Победа игрока О)

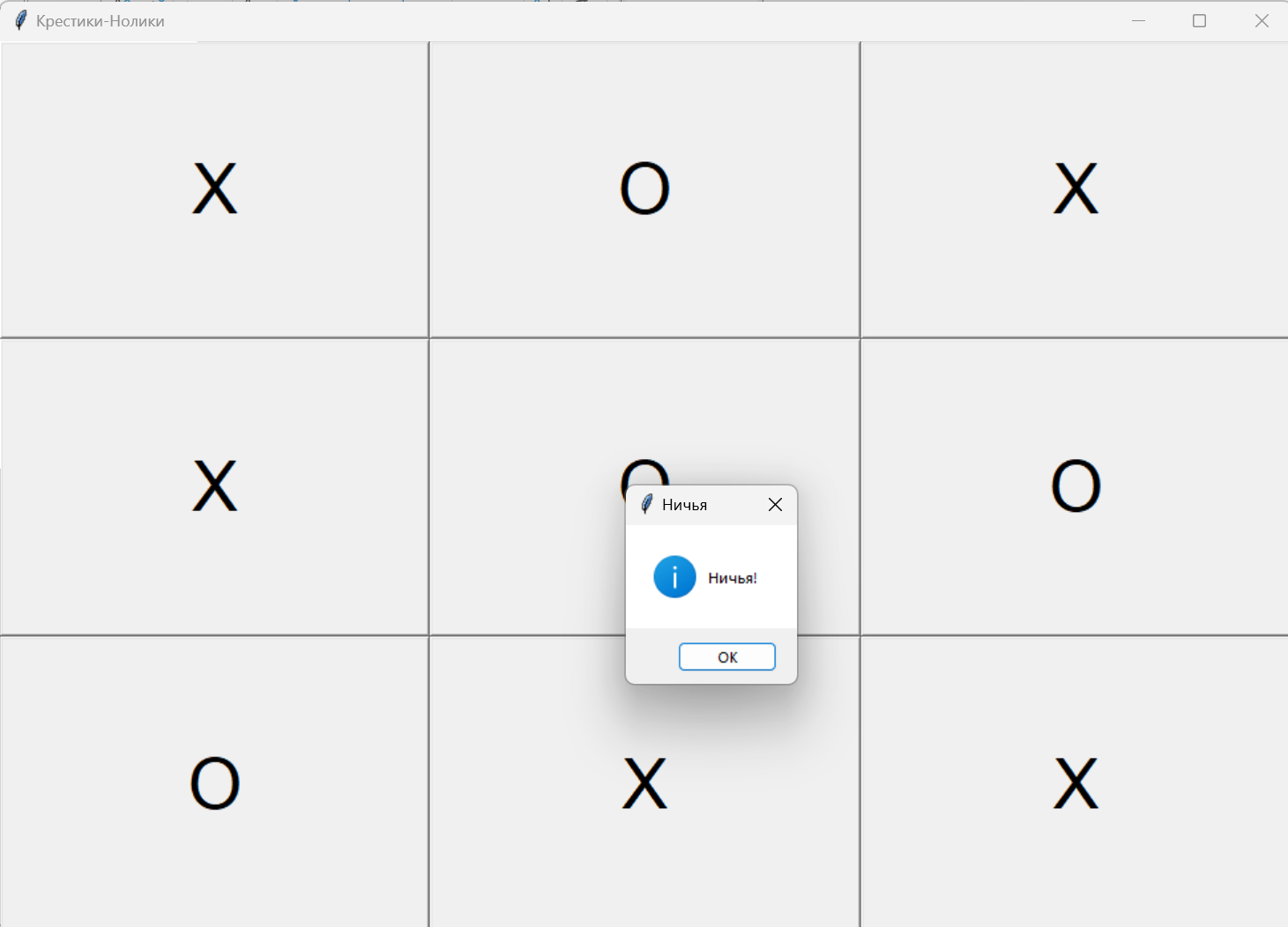


Рис. 4 (Ничья)

# Заключение

В данном проекте была создана и описана интерактивная графическая игра "Крестики-Нолики" с использованием библиотеки Tkinter. Основное внимание было уделено правильности отображения игрового поля, реализации логики хода игроков и проверки условий победы и ничьей. Реализованная игра позволяет пользователю легко взаимодействовать с игровым интерфейсом и наслаждаться классической игрой в удобном формате. Полученный опыт может быть полезен для дальнейших проектов в области разработки графических интерфейсов и игр.

# Список использованных источников

1. tkinter — Python interface to Tcl/Tk: [Электронный ресурс]. https://docs.python.org/3/library/tkinter.html (Дата обращения: 29.05.2024)
2. Русскоязычная документация по Python [Электронный ресурс]. https://pylessons.readthedocs.io/ru/latest/ (Дата обращения: 29.05.2024)
3. Правила игры в крестики-нолики Python [Электронный ресурс]. <https://tictactoefree.com/ru/pravila> (Дата обращения: 29.05.2024)