**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Измерительно-вычислительные комплексы»

**Лабораторная работа**

**По дисциплине «Алгоритмы и структуры данных»**

**Тема Лабораторная работа №10**

|  |  |
| --- | --- |
| **Подп. и** |  |
| **Инв** |  |
| **Вза** |  |
| **Подп. и** |  |
| **Инв.** |  |

**Пояснительная записка**

**дата**

Р.02069337. <23/710>-<04> ПЗ-<2-зн. номер

редакции> Листов 7

**м. инв. №**

**. № дубл.**

**Исполнитель**: студент гр. ИСТбд-23

**дата**

*Калашников А. А.*

« 20 » мая 2025 г.

2025

**№ подл.**

**Введение**

В данном проекте разрабатывается приложение для игры "Крестики-нолики". В приложении реализована функциональность для управления игровым процессом, проверки ходов и определения победителя. Используемый подход основан на объектно-ориентированном программировании с применением библиотеки Tkinter для графического интерфейса.

1. **Проектная часть**
   1. **Постановка задачи на разработку приложения**

Задача разработки заключается в создании интерактивного приложения, которое позволяет игрокам вести игру "Крестики-нолики". Приложение должно обеспечивать возможность установки крестиков, проверку условий выигрыша и корректность ходов.

* 1. **Математические методы**

Применяемая математическая модель для определения состояния игры основывается на анализе текущего состояния доски и выборе лучших ходов для игрока и ИИ. Выбор этого подхода обусловлен необходимостью быстрой оценки возможностей для победы и предотвращения проигрыша.

* 1. **Архитектура и алгоритмы**
     1. **Архитектура**

Архитектура приложения включает в себя следующие основные структуры данных и функции:

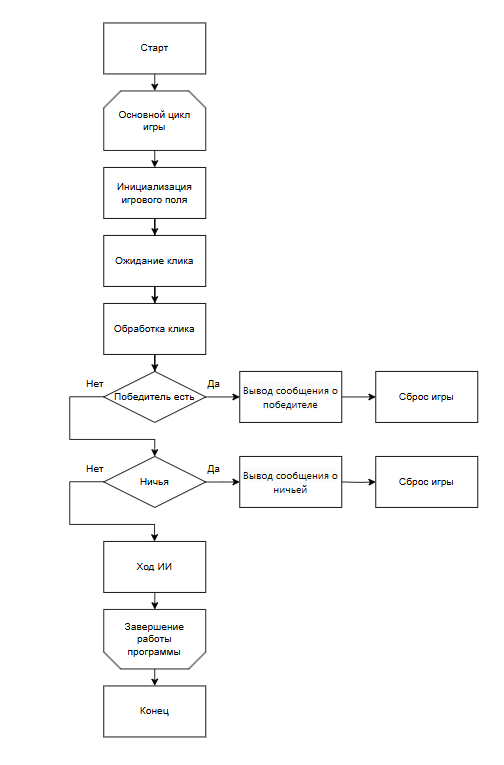
* Структуры данных:
  + board: список, представляющий состояние игрового поля (размер 3×3).
  + human\_player: переменная, хранящая игрока (X или O)
  + current\_player: переменная, хранящая текущий ход (X или O)
  + win\_combinations: Все выигрышные комбинации
* Функции:
  + create\_board: создает игровое поле.
  + on\_button\_click: Обработка хода.
  + computer\_move(): выполнение хода ИИ.
  + check\_winner\_board(): определяет победителя.
  + reset\_board(): сброс состояния игры для новой партии.
  + minimax(): вычисление лучшего хода для ИИ.
    1. **Алгоритм обработки данных**

Пример алгоритма, который выполняется при нажатии игроком на клетку:

1. Запрос номера клетки для установки фишки.
2. Проверка возможности хода (если клетка свободна).
3. Если ход возможен:
   * Обновить состояние доски.
   * Проверить на наличие победителя.
   * Переключить игрока (между X и O) для следующего хода.
     1. **Алгоритм игры**

Этот алгоритм обеспечивает основные действия в игре и включает следующие шаги:

1. Инициализация игрового поля.
2. Ожидание хода от игрока или ИИ.
3. Проверка состояния игры после каждого хода.
4. Определение победителя или ничьей по окончании игры.
   1. **Тестирование**

****

**1.1.1. Описание отчета о тестировании:**

Этот отчет описывает результаты тестирования игры "Крестики-нолики", реализованной с помощью Python и библиотеки Tkinter. Тестирование охватывало функциональность игрового процесса, включая проверку хода игры, определения победителя и обработки ничьих.

**1.1.2. Цель тестирования:**

Цель тестирования заключалась в проверке корректности работы всех функций игры: запуск игры, регистрация ходов игроков (включая ход искусственного интеллекта), определение победителя или ничьей, и корректное отображение игрового поля.

**1.1.3. Методика тестирования:**

Тестирование проводилось путем ручного прохождения различных игровых сценариев, включая различные комбинации ходов игроков и проверки реакции игры на них. Использовалось ручное тестирование для проверки всех возможных вариантов игры.

**1.1.4. Проведенные тесты:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Тестовый случай** | **Описание** | **Ожидаемый результат** | **Фактический результат** | **Статус** |
| ТС1 | Корректность смены хода между игроками "X" и "O". | Ходы корректно сменяются между игроками | Ходы успешно и корректно сменяются между ходами игроков | Пройден |
| ТС2 | Корректное определение победителя при наличии выигрышной комбинации | Победитель определяется корректно при наличии выигрышной комбинации | Победитель успешно и корректно определяется | Пройден |
| ТС3 | Корректное определение ничьей при заполнении всего игрового поля без выигрышной комбинации | При заполнении всего игрового поля без выигрышной комбинации программа выводит сообщение о ничьей | Программа успешно выводит сообщение о ничьей | Пройден |
| ТС4 | Корректность работы алгоритма искусственного интеллекта | Искусственный интеллект работает корректно, не допускает ошибок и неэффективных ходов при игре | Искусственный интеллект не допускает ошибок и неэффективных ходов при игре | Пройден |
| ТС5 | Проверка отсутствия ошибок и исключений при работе приложения | Приложение работает корректно и выполняет свои функции | Приложение работает корректно и выполняет свои функции | Пройден |

**1.1.5. Выводы:**

На основе проведенных тестов можно заключить, что игра "Крестики-нолики" работает корректно. Алгоритм определения победителя и ничьей функционирует без ошибок. Искусственный интеллект демонстрирует разумную стратегию игры. Графический интерфейс интуитивно понятен и удобен в использовании. Приложение стабильно работает и не выдает ошибок.

1. **Источники, использованные при разработке**

https://metanit.com/python/tkinter/ Руководство по Tkinter