МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВЕРО-ВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ М.К.

АММОСОВА»

ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАТИКИ

**Отчёт по учебной ознакомительной практике**

по дисциплине «Математическое моделирование и вычислительная математика» направления 01.03.02 «Прикладная математика и информатика»

студента группы ИИ-ПМИ-23 Блахирова Тимура Викторовича / \_\_\_\_\_\_\_\_\_ (подпись) Руководитель практики: доцент, к.т.н. Акимов М.П.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_

(оценка / подпись)

Якутск 2025

## СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc5219)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc5220)

[ЗАКЛЮЧЕНИЕ 7](#_Toc5221)

[ПРИЛОЖЕНИЯ 8](#_Toc5222)

# ВВЕДЕНИЕ

Тему для проекта по учебной практике нам студентам дали возможность выбрать самим, что ближе лежит, главное чтобы тема была связана с IT, для достижения целей надо было пройти онлайн-курсы, а так же выложить все файлы связанные с практикой на платформу GitHub.

Я выбрал мини-проект «Крестики-нолики»: это приложение – игра основанный на языке Python 3.9.3

В приложении можно попытаться переиграть компьютера, в другую очередь компьютер будет делать все, чтобы не проиграть вам.

В приложении будет как консольный вид, так и графический, и там и там, при входе будет видна доска с размером 3x3, где будут происходить бои.

Я выбрал именно этот проект, потому что я с детства интересуюсь играми, и мне стало интересно сможет ли выиграть человек который знает все шаблоны ходов для победы, или же компьютер.

# ОСНОВНАЯ ЧАСТЬ

Для начало нужно было начать ввести дневник и параллельно начать делать отчет практики, для этого я ознакомился с примерами отчета, ГОСТ Р 7.0.97-2025 и ГОСТ 7.32-2017.

Дальше надо было выбрать подходящий онлайн-курс для реализации моего проекта, так как мне нужен был AI\*, я выбрал курс Машинное обучения, к сожалению мне он не пригодился, но т.к моя учеба связана с ним, лишним не будет. А также я частично прошел курс по языку программирования Python, чтобы освежить память.

Обучение на курсе в общем продлилось 7 дней. Далее прикреплю сертификат об прохождении курса и прогрессу.

Затем я стал изучать объектно-ориентированное программирование (ООП) на Python и постепенно начинать сам проект. Изучал я его через официальную документацию Python и различные онлайн-ресурсы. ООП предназначено для того, чтобы удобно структурировать код и создавать сложные системы с помощью классов и объектов. Выяснилось, что этот подход позволяет создавать легко расширяемые и поддерживаемые приложения.

Программа была написана в среде разработки на Python 3.9 с использованием встроенных библиотек. Был выбран стэк Python/консольное приложение, так как это позволяло сосредоточиться на логике игры без отвлечения на графический интерфейс. Использовалась встроенная библиотека random для реализации логики искусственного интеллекта.

При написании программы я первым делом думал над архитектурой приложения: как организовать игровую логику, разделить ответственность между компонентами. В итоге я выбрал объектно-ориентированный подход с классом Game, который инкапсулирует состояние игрового поля.

Из-за того, что приложение было консольным, возникла проблема: нужно было придумать удобный способ отображения игрового поля и ввода данных пользователем. Поэтому мой выбор пал на пошаговый вывод поля с нумерацией строк и столбцов.

Пройдусь по основным компонентам программы. Класс Game отвечает за состояние игрового поля и его отображение. Метод display\_board() может показывать номера строк и столбцов в зависимости от параметра numbers, что помогает пользователю ориентироваться при вводе координат.

Функция check\_win() содержит логику проверки всех возможных выигрышных комбинаций: горизонтали, вертикали и диагонали. Также она проверяет условие ничьи, когда все клетки заполнены, но нет победителя.

Особого внимания заслуживает подобия ИИ в функции computerMove(). Он использует двухуровневую логику: сначала проверяет, может ли компьютер выиграть следующим ходом, затем — может ли блокировать выигрыш игрока. Если ни то, ни другое невозможно, псевдо ИИ делает случайный ход в свободную клетку.

Главный игровой цикл организован через while True, внутри которого поочередно вызываются ходы для 'x' (игрок) и 'o' (компьютер). После каждого хода проверяется состояние игры, и при обнаружении победы или ничьи предлагается сыграть заново.

Во время разработки возникло несколько проблем. Во-первых, нужно было корректно обрабатывать ввод пользователя — обеспечить, чтобы координаты были в допустимом диапазоне и клетка была свободна. Для этого была реализована рекурсивная функция play(), которая вызывается повторно при некорректном вводе.

Во-вторых, возникла сложность с проверкой выигрышных комбинаций. Изначально функция all\_equal() была реализована как лямбда-функция, но в процессе тестирования выяснилось, что нужно дополнительно проверять, что клетки не пустые (не None).

В-третьих, логика ИИ требовала тщательной отладки. Первоначальная версия могла пропускать выигрышные ходы или, наоборот, делать неоптимальные ходы. После нескольких итераций доработки алгоритм стал достаточно умным, чтобы никогда не проигрывать при правильной игре.

Несмотря на кажущуюся простоту игры, реализация потребовала внимания к деталям и тщательного тестирования граничных случаев, таких как ничья, победа в последний момент и обработка некорректного ввода пользователя.

# ЗАКЛЮЧЕНИЕ

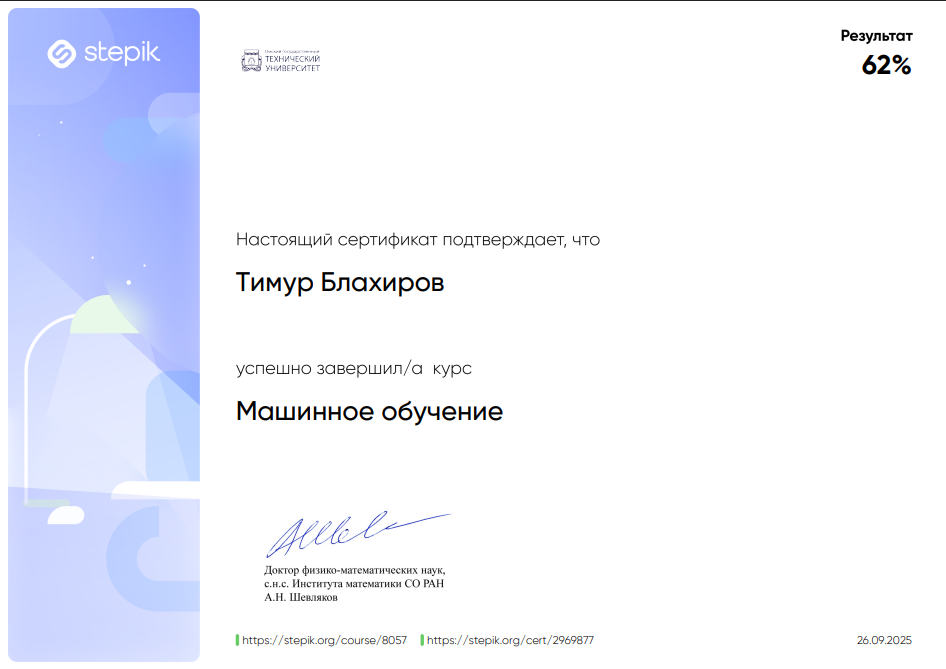
В итоге получилось игра “Крестики-нолики”, в котором можно:

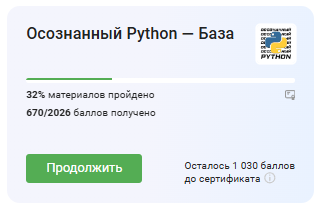
* Играть против компьютера с трудной логикой;
* Делать ходы по координатам, выбирая позицию на поле;
* Наблюдать за стратегией компьютера, который пытается выиграть или блокировать ходы игрока;
* Определять победителя;

В ходе работы и подготовки, я подкрепил свои знания по машинному обучению, а также освежил свои навыки программирования на Python, прошел онлайн - курс на платформе Stepik, разработал консольное приложение\*, где полностью работают функции.

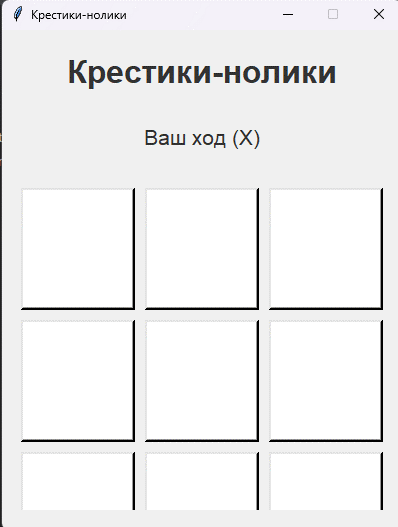
# ПРИЛОЖЕНИЯ

**Снимки экрана**

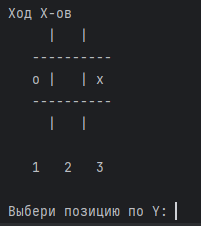


*Рисунок 1. Сертификат курса по машинному обучению*

*Рисунок 2. Прогресс курса по Python*



*Рисунок 3. Графический вид игры*

**

*Рисунок 4. Консольный вид игры*

## Код проекта

<https://github.com/svyyyx/SummerPractice> - ссылка на репозиторий GitHub

**Использованные онлайн-ресурсы**

* stepik.org
* youtube.org
* python.org
* stackoverflow.com