МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | |  | | ТЕХНИЧЕСКОЕ ЗАДАНИЕ  на лабораторную работу №9  по дисциплине «Алгоритмы и структуры данных»  Тема «Игра крестики-нолики» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-21 Мишин А.В.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**1. Введение**

Данное техническое задание посвящено разработке компьютерной программы "Крестики-нолики", которая позволяет пользователям играть в классическую игру. Игра предусматривает два режима: игра против другого игрока и игра против компьютера. Программа должна иметь графический интерфейс для удобного взаимодействия с пользователем.

**2. Основания для разработки**

Разработка программы осуществляется в рамках учебного курса по дисциплине "Алгоритмы и структуры данных".

**3. Назначение разработки**

Программа предназначена для реализации игры "Крестики-нолики" с возможностью выбора режима игры, отображения текущего состояния игры, а также определения победителя. Основные функции программы включают:

* Создание игрового поля.
* Выбор режима игры: против одного или двух игроков.
* Логика игры для человека и компьютера.
* Отображение информации о ходе игры, победителе и времени игры.
* Возможность перезапуска игры.

**4. Требования к программе**

**4.1 Требования к функциональным характеристикам**

**4.1.1 Программа должна обеспечивать выполнение следующих функций:**

* Реализация графического интерфейса с использованием библиотеки tkinter.
* Реализация игрового поля размером 3x3.
* Обработка хода игроков (включая компьютер).
* Определение победителя (или ничьей).
* Подсчет времени игры с отображением на экране.
* Переключение хода между игроками.
* Возможность перезапуска игры.

**4.1.2 Организация входных и выходных данных**

Входные данные: выбор режима игры, действия игроков (клики на клетках поля).

Выходные данные: отображение состояния игры (текущий ход, победитель или ничья, время игры).

**4.2 Требования к надежности**

Программа должна обеспечивать стабильную работу, корректную обработку ввода пользователя и надежно отображать результаты игры.

**4.3 Требования к составу и параметрам технических средств**

Программа должна функционировать на персональных компьютерах с операционной системой Windows, Linux и macOS, поддерживающих Python версии 3.x.

**4.4 Требования к информационной и программной совместимости**

Программа должна быть совместима с операционными системами Windows, Linux и macOS.

Программа должна быть написана на Python с использованием библиотеки tkinter для создания графического интерфейса.

**4.5 Условия и срок хранения**

Обеспечение свободного доступа к проекту в репозитории до окончания учебы.

**5. Требования к программной документации**

В состав документации должны входить:

* Техническое задание.
* Руководство пользователя.
* Руководство программиста.
* Пояснительная записка.
* Тестовая документация.

**6. Стадии и этапы разработки**

1. Анализ требований.
2. Проектирование архитектуры программы.
3. Разработка пользовательского интерфейса.
4. Реализация логики игры.
5. Реализация алгоритма для игры против компьютера.
6. Реализация функций подсчета времени игры.
7. Тестирование и отладка.
8. Документирование.
9. Внедрение (сдача проекта).

**6.1 Распределение обязанностей в команде разработчиков**

* Дизайнер: Мишин Артур;
* Разработка логики работы приложения: Мишин Артур;
* Тестировщик/QA: Мишин Артур;
* Документация: Мишин Артур

**7. Порядок контроля и приемки**

7.1. **Контроль осуществляется посредством проведения тестов, проверяющих функциональность и надежность программы по следующим критериям:**

* Правильная обработка хода игроков.
* Корректное отображение состояния игры.
* Проверка логики игры против компьютера.
* Подсчет времени игры.
* Стабильность работы пользовательского интерфейса.

**7.2. Приемка проекта осуществляется в рамках лабораторных работ, где проводится повторное тестирование и оценка работы.**