МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

Курсовая работа

по дисциплине «Алгоритмы и структуры данных» Тема «Компьютерная игра эндшпиль «Король, 2 коня - Король, конь, пешка»»

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

Руководство программиста

Р.02069337.<23/710 >-<18> РП-<2-зн. номерредакции> Листов (7)

Исполнитель:

студент гр. ИСТбд-23

*Шабаев Р.И*

« » 2025 г.

2025

# Назначение и условия применения программы

* 1. **Назначение и функции, выполняемые приложением**

Программа представляет собой игровое приложение, реализующее игру в шахматы. В игре предусмотрены стандартные правила передвижения и захвата шахматных фигур, а также правила игры.

Основные функции программы включают:

* + - Инициализация игрового поля и фигур.
    - Визуализация игрового процесса.
    - Обработка ходов игроков, включая движение и захват фигур.
    - Определение победителя.
    - Авторизация пользователя.

# Условия, необходимые для использования приложения

Для работы приложения необходимы следующие условия:

* + - Операционная система: Windows, macOS, Linux.
    - Инструментальная среда: Python 3.x.
    - Необходимые библиотеки:
    - Pygame: для реализации графического интерфейса и обработки ввода с клавиатуры.
    - Python стандартные библиотеки: для работы с файловой системой, а также реализации логики игры.

# Характеристики программы

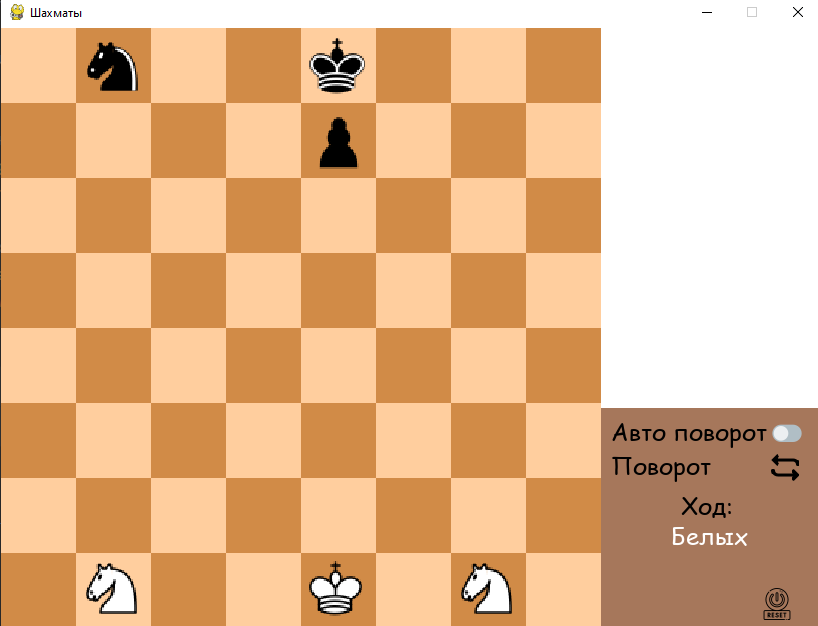
* 1. **Характеристики приложения**

Объем кода**:** 432 строк Кол-во алгоритмов: 5

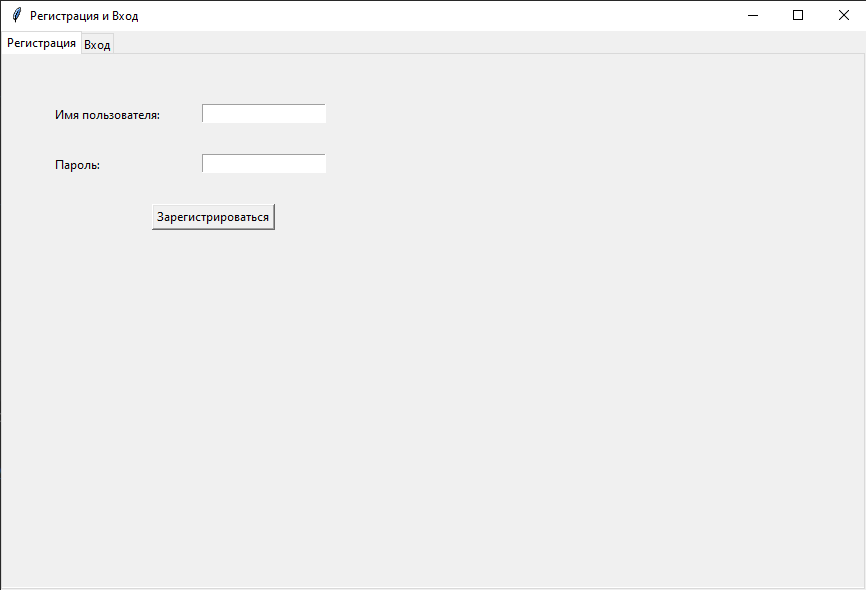
Используемые библиотеки: Pygame, стандартные библиотеки Python: time, tkinter.

Приложение представляет собой окончание игры в шахматы(эндшпиль) с элементами авторизации и регистрации пользователей. Вот основные этапы работы приложения:

Главное окно игры:



Авторизация и регистрация:



# Особенности реализации приложения

Для реализации игры использованы следующие структуры данных:

* + - **Массивы** для представления игрового поля. Каждый элемент массива хранит информацию о состоянии клетки (пусто, белая фигура, черная фигура).
    - **Списки** для хранения позиций фигур и очередности хода.
    - **Классы** для клеток и фигур. В каждом классе расписана логика каждой фигуры.

# Обращение к программе

Используемые библиотеки:

* **pygame**: Это библиотека для создания игр, которая предоставляет возможности для работы с графикой, звуком, событиями ввода (клавиши, мышь).
* **time**: Библиотека Python, которая используется для работы с временем. Описание функций и методов программы:
* Вот компактное описание функций и методов без пустых строк:
* Функции аутентификации:
* generate\_key() - Генерирует криптографический ключ для шифрования паролей
* save\_key(key) - Сохраняет ключ шифрования в файл "secret.key"
* load\_key() - Загружает ключ шифрования из файла
* encrypt\_password(password) - Шифрует пароль с использованием Fernet
* decrypt\_password(encrypted\_password) - Расшифровывает пароль
* save\_credentials(username, password) - Сохраняет логин и зашифрованный пароль в файл
* check\_credentials(username, password) - Проверяет правильность учетных данных
* Графические функции:
* draw\_screen() - Отрисовывает игровое поле, фигуры, доступные ходы и интерфейс
* draw\_pieces() - Рисует шахматные фигуры на доске
* draw\_options() - Отрисовывает элементы управления (поворот доски, сброс)
* draw\_winner() - Показывает сообщение о победе или ничьей
* draw\_available\_moves() - Отмечает доступные для хода клетки
* draw\_current\_player() - Показывает, чей сейчас ход
* draw\_promote\_options() - Отображает выбор фигуры при превращении пешки
* Игровая логика:
* validate\_chosen\_piece() - Проверяет, можно ли выбрать указанную фигуру
* get\_piece\_moves() - Возвращает список допустимых ходов для фигуры
* validate\_target\_piece() - Проверяет, можно ли сделать ход в указанную клетку
* promotion() - Обрабатывает выбор фигуры при превращении пешки
* move() - Выполняет ход фигурой, включая специальные правила (рокировка, превращение)
* start() - Основной игровой цикл
* board.py (Логика шахматной доски) Основные методы класса Board:
* \_\_init\_\_() - Инициализирует доску и расставляет фигуры
* get\_piece(pos) - Возвращает фигуру по указанной позиции
* match\_color() - Проверяет совпадение цвета фигуры с заданным
* pawn\_move() - Обрабатывает специальные правила хода пешки (взятие на проходе)
* king\_move() - Обрабатывает правила хода короля (включая рокировку)
* get\_valid\_moves() - Возвращает допустимые ходы для фигуры
* update() - Обновляет состояние доски после хода
* can\_check() - Проверяет, объявляет ли фигура шах
* get\_king\_pos() - Находит позицию короля указанного цвета
* stalemate() - Проверяет условия пат
* get\_checked\_when\_move() - Проверяет, окажется ли король под шахом после хода
* Вспомогательные функции:
* generate\_cases() - Создает копию доски для симуляции ходов
* get\_row\_col() - Преобразует шахматную нотацию в координаты доски
* get\_square\_name() - Преобразует координаты доски в шахматную нотацию
* pieces.py (Классы шахматных фигур) Базовый класс Piece:
* \_\_init\_\_() - Инициализирует фигуру с позицией и цветом
* square\_in\_board() - Проверяет, находится ли клетка на доске
* can\_move() - Абстрактный метод (реализуется в дочерних классах)
* Классы фигур (реализуют can\_move()):
* Rook - Ладья (ходы по вертикали/горизонтали)
* Bishop - Слон (ходы по диагонали)
* Queen - Ферзь (комбинация ладьи и слона)
* King - Король (ход на 1 клетку в любом направлении + рокировка)
* Knight - Конь (ход буквой "Г")
* Pawn - Пешка (специальные правила: взятие по диагонали, превращение)

# Сообщения

## Ошибка при вводе данных:

* "Ошибка: Пустое поле логина или пароля!"
* "Неправильный логин или пароль"

## Подтверждения:

* "Регистрация успешна!"
* "Вход выполнен успешно!"