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»**

# **РУКОВОДСТВО ПРОГРАММИСТА**

**Тема:** **Компьютерная игра эндшпиль «Король, 2 пешки -Король, пешка»**

Р.02069337.22/257-12 РП-01

Листов 18

**Исполнитель**:

студент гр. ИСТбд-31

*Кузьмин Владимир Сергеевич*

« » 2025 г.

**2025**

**1. Назначение и условия применения программы**

**1.1 Назначение и функции, выполняемые приложением**

Приложение «Эндшпиль, король и пешки» реализовано для развлечения пользователей. Приложение реализовано по следующим правилам:

**Поле и игроки.** Игра ведется между двумя соперниками: пользователем и компьютером. Игровое поле представляет собой стандартную шахматную доску размером 8×8 клеток. Игрок управляет белыми фигурами, а компьютер — черными. Фигуры перемещаются по стандартным шахматным правилам.

**Порядок ходов.** Игроки ходят по очереди. Первый ход делает игрок за белые фигуры.

**Цель игры.** Срубить вражеского короля раньше противника. Игрок. Последний оставшийся с королем на доске - побеждает.

В приложении предоставляется возможность авторизоваться и входить в свой аккаунт. В приложении реализованы правила игры в шахматы - ходы фигур, превращение пешек в ферзи, рубка фигур.

**1.2 Условия, необходимые для использования приложения**

Рекомендуется к использованию на Windows 10.

При создании программы используются встроенные библиотеки “os”, “random”, “json”, “hashlib” и сторонние библиотеки “tkinter 8.6”, “tkinter.messagebox”, “pycryptodome” (модули Crypto.Cipher, Crypto.Util.Padding, Crypto.Random), а также “base64” для кодирования данных.

Разработка велась в Visual Studio Code 1.98.0. на версии языка программирования Python 3.13.2

**2. Характеристики программы**

**2.1 Характеристики приложения**

Значимых строк кода 515

Структура данных одна – массив.

**Использованные библиотеки:**

**“os”** – стандартная библиотека для работы с операционной системой. Позволяет управлять файлами и директориями, получать доступ к переменным окружения и выполнять системные команды.

**“tkinter”** – встроенная библиотека для создания графического интерфейса. Позволяет создавать окна, кнопки, поля ввода и другие элементы GUI.

**“messagebox (tkinter)”** – модуль для вывода всплывающих окон с сообщениями, предупреждениями и запросами подтверждения.

**“Json"** – модуль для работы с JSON-форматом, позволяющий сохранять и загружать данные в файлы или передавать их между программами.

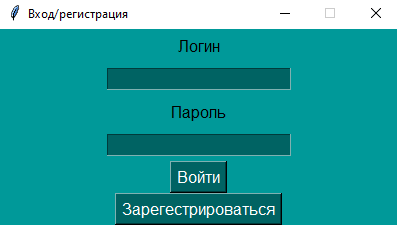
**“random”** – библиотека для генерации случайных чисел и значений. Используется, например, при выборе случайных позиций на доске.

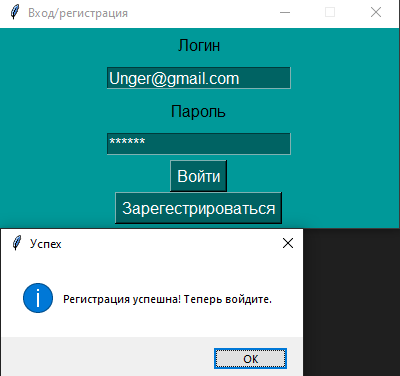
**“Crypto.Cipher (AES)”** – модуль из библиотеки PyCryptodome, предоставляющий реализацию алгоритма шифрования AES. Используется для безопасного хранения паролей.

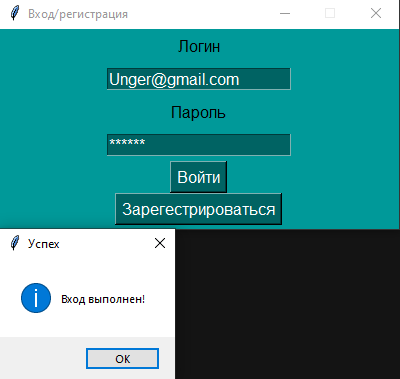
**“Crypto.Util.Padding”** – вспомогательный модуль для дополнения (padding) данных перед шифрованием.

**“Crypto.Random”** – модуль для генерации криптографически стойких случайных данных, например, ключей шифрования.

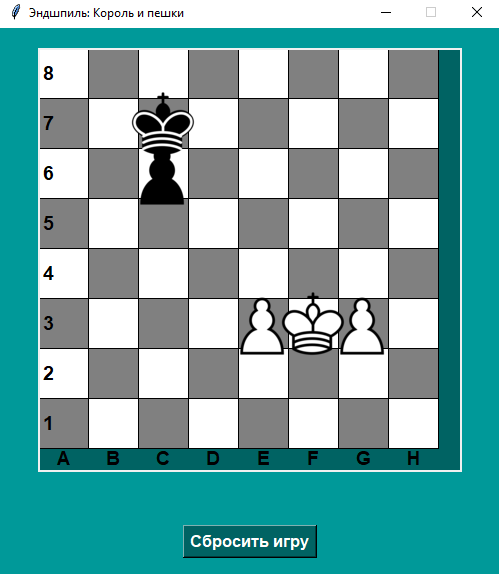
**“Base64"** – библиотека для кодирования и декодирования данных в формате Base64, используется при шифровании паролей.

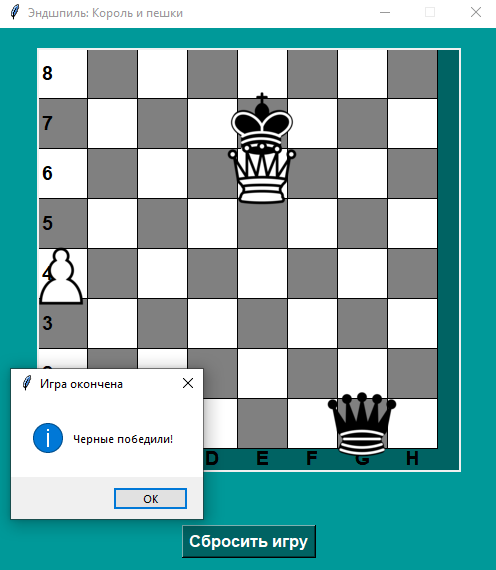
Работа приложения начинается с окна авторизации и регистрации: 

Успешная регистрация:  


Успешная авторизация:  


Отрисовка игрового поля:



Конец игры:  


**2.2 Особенности реализации приложения**

В программе в качестве структуры данных используется массив, который применяется для отрисовки игрового поля и проверки ходов. Данный подход позволяет легко отслеживать, какие клетки заняты, а какие свободны. Кроме того, двумерный массив упрощает проверку условий победы, так как позволяет эффективно анализировать расположение фигур в различных направлениях.

**3. Обращение к программе**

Функции:  
1) load\_or\_create\_key()

Проверяет существование файла с ключом шифрования. Если файл есть, загружает ключ, если нет — создаёт новый 16-байтовый ключ AES-128, сохраняет его в файл и возвращает.

2) encrypt\_password(password)

Шифрует пароль с использованием алгоритма AES. Она генерирует случайный ключ, шифрует пароль и возвращает результат в формате IV:зашифрованный\_пароль.

3) decrypt\_password(encrypted\_password)

Расшифровывает пароль с помощью AES

4) load\_users()

Загружает список пользователей из файла, если файл существует. Если файл не найден, она возвращает пустой список.

5) save\_users(users)

Сохраняет пользователя в файл

6) register()

Обрабатывает процесс регистрации пользователя. Она проверяет корректность введённых данных, такие как логин и пароль, и убеждается, что они соответствуют определённым условиям. Если данные корректны и логин уникален, функция сохраняет информацию о пользователе и сообщает о успешной регистрации.

7) login()

Осуществляет процесс авторизации пользователя. Она проверяет, чтобы все поля были заполнены, затем ищет пользователя в базе данных. Если логин найден, она проверяет правильность пароля, используя функцию для расшифровки. При успешной авторизации запускается игра, в противном случае выводится сообщение об ошибке.

Класс ChessGame (Служит для работы с игровой логикой и обработке различных событий игрового характера)

1) \_\_init\_\_(self, master)

Конструктор метода.

2) init\_board(self)

Создает шахматную доску, устанавливает ее размер, рисует клетки с чередующимися цветами и добавляет координаты по краям доски. Клетки рисуются на канвасе, и для каждой строки и столбца добавляются соответствующие текстовые метки (цифры для строк и буквы для столбцов).

3) get\_random\_empty\_position(for\_pawn=False, is\_white=False, is\_king=False)

Выбирает случайную пустую клетку на доске, с учетом некоторых условий. Она проверяет, чтобы выбранная позиция была свободной, и в зависимости от переданных флагов, учитывает дополнительные ограничения, такие как запрет на размещение пешек в первых или последних рядах, или ограничение для королей на определенные строки доски.

4) draw\_pieces(self)

Отвечает за отрисовку фигур на доске. Она удаляет все текущие изображения фигур на холсте, затем загружает изображения для разных типов фигур (король, ферзь, пешка) для черных и белых. Для каждой клетки доски, в зависимости от содержимого, она отображает соответствующую фигуру на холсте с помощью изображений.

5) is\_king\_captured(self, player)

Проверяет, захвачен ли король указанного игрока

6) on\_click(self, event)

Обрабатываем клик по доске

7) move\_piece(self, from\_pos, to\_pos)

Перемещает фигуру и сразу проверяет победу

8) is\_valid\_move(self, piece, from\_row, from\_col, to\_row, to\_col)

Метод проверяет, является ли ход допустимым для выбранной фигуры. Он учитывает тип фигуры, ее правила движения и наличие преград на пути (например, других фигур).

9) reset\_game(self)

Сбрасывает игру, создавая новую доску

10) show\_victory\_message(self, message)

Показывает всплывающее окно с сообщением о победе и перезапускает игру

11) check\_victory(self)

Метод проверяет, есть ли на доске короли обоих игроков. Если одного из королей нет, игра заканчивается, и объявляется победитель.

12) get\_all\_valid\_moves(self, piece, row, col)

Возвращает все допустимые ходы для данной фигуры

13) is\_check(self, color)

Проверяет, находится ли король на текущем цвете под шахом.

14) find\_king(self, color)

Находит позицию короля на поле для данного цвета.

15) is\_safe\_move(self, row, col)

Проверяет, безопасен ли ход на данную клетку для короля.

16) is\_under\_attack(self, row, col)

Проверяет, атакует ли клетку вражеская фигура.

17) king\_escape(self)

Метод проверяет, может ли король избежать шаха, передвигаясь на безопасные клетки. Если существует безопасный ход, король перемещается, и проверяется, остаётся ли он под шахом. Если удаётся избежать шаха, функция возвращает True, иначе — False.

18) simulate\_move(self, from\_row, from\_col, to\_row, to\_col)

Временный ход для проверки безопасности клетки

19) undo\_move(self, from\_row, from\_col, to\_row, to\_col)

Откат хода после проверки безопасности клетки

20) ai\_move(self)

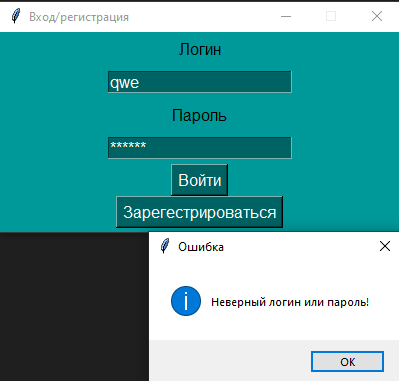
Метод управляет ходами ИИ, при этом она делает выбор по определенным приоритетам. Сначала он проверяет, можно ли сразу взять короля противника. Если король под шахом, ИИ пытается найти безопасный ход или ход с захватом. Далее ИИ проверяет, может ли он атаковать фигуры противника, делает ход ферзя или продвигает пешку, если это возможно. После этого ИИ выбирает обычный ход для других фигур. Ход выполняется в зависимости от приоритета, начиная с самых критичных ситуаций, таких как захват короля.

21) start\_game()

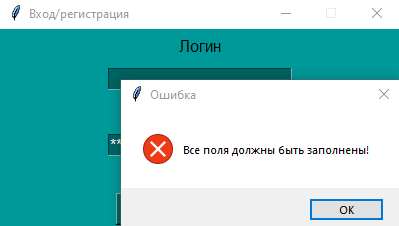
Запуск игры

**4. Сообщения**

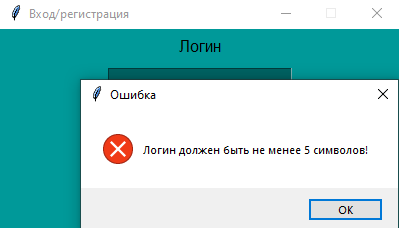
Пользователь не найден или введен неправильный пароль:



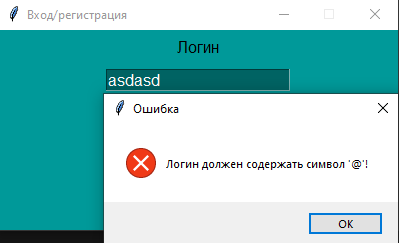
Какое-либо из полей было не заполнено:



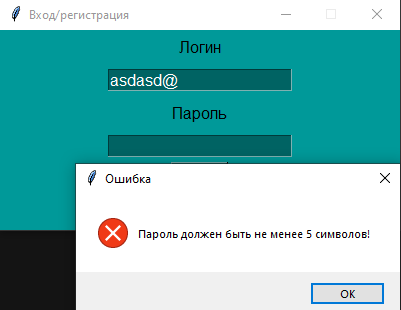
Ограничение на минимальную длину логина:



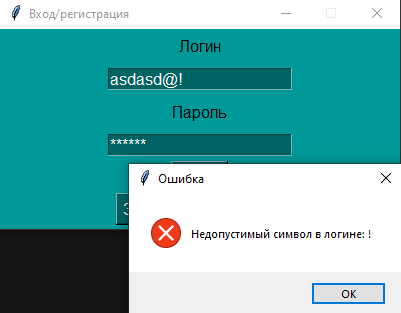
Ограничение требующее наличие знака “@” в логине:

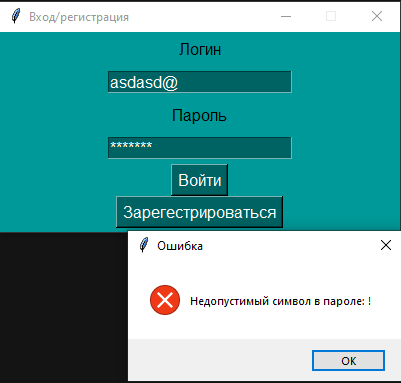


Ограничение на минимальную длину пароля:

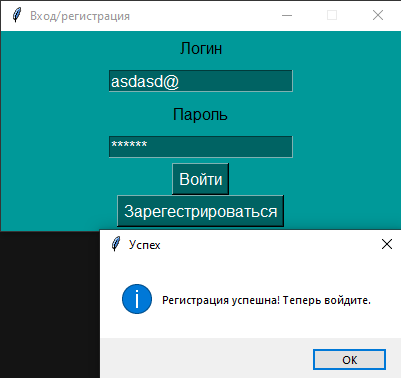


Ограничение на недопустимые символы в логине:

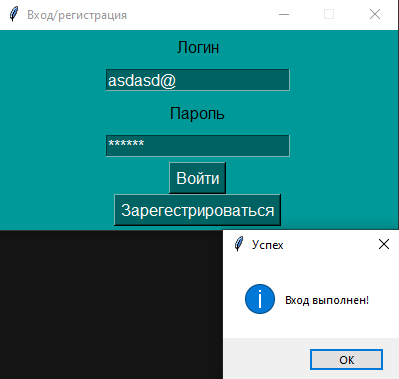


Ограничение на недопустимые символы в пароле:  


Сообщение об успешной регистрации:



Сообщение об успешной авторизации:



Сообщение об победе определенной стороны:

