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.** |  |

Пояснительная записка

Р.02069337.<23/710 >-<18> ПЗ-<2-зн. номерредакции> Листов (8)

Исполнитель:

студент гр. ИСТбд-23

*Шабаев Р.И*

« » 2025 г.

2025

# Введение

Наименование приложения:

Компьютерная логическая игра «Шахматы(эндшпиль)» Условное обозначение:

"Chess(endgame)"

Реализована на языке Python с использованием библиотеки Pygame. Данная игра представляет собой модификацию классических шахмат.

# Проектная часть

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

Определяется заданием на курсовую работу

# Математические методы

Для авторизации используется метод симметричного шифрования данных. Логин и пароль пользователя кодируются с помощью Unicode каждого символа

умноженного на два, а для дешифровки применяется обратный процесс.

Для игры в шахматы применяются стандартные математические методы, описывающие движение и захват фигур на доске. Каждый ход реализован в виде координатных преобразований, в которых учитывается возможность захвата противника, а также продвижение фигур до конца доски.

# Архитектура и алгоритмы

* + 1. **Архитектура**

Приложение имеет модульную архитектуру, состоящую из следующих компонентов:

* + - * **Регистрация и вход (regis)** — окно для ввода логина и пароля с возможностью их регистрации или авторизации.
      * **Игра (main)** — сама игровая логика, включающая доску, фигуры, а также кнопку restart.
      * **Шифрование данных (encode\_data и decode\_data)** — функции для кодирования и декодирования данных пользователя.

# 1.3.3. Алгоритм игры.

Алгоритм игры включает следующие этапы:

1. **Создание доски** — генерируется начальная шахматная доска с расставленными фигурами для двух игроков.
2. **Выбор фигуры** — игрок выбирает фигуру для перемещения, на экране отображаются возможные ходы.
3. **Выполнение хода** — после выбора клетки для перемещения фигура двигается, если ход возможен.
4. **Превращение фигуры** — если пешка достигает противоположной стороны доски, она превращается в любую фигуру, кроме короля.
5. **Проверка победителя** — игра завершается, когда все фигуры одного игрока уничтожены.

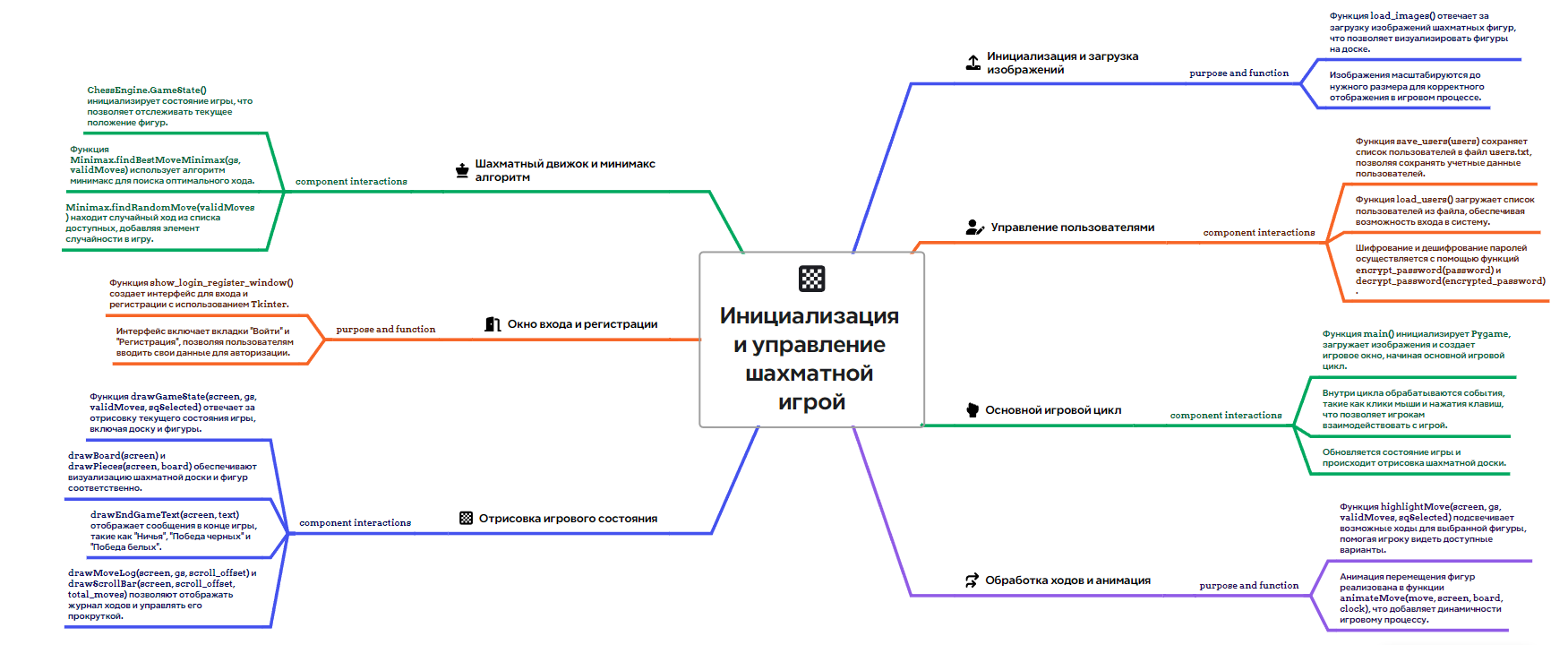
# 1.3.2. Алгоритм шифрования и дешифрования данных

Алгоритм шифрования данных (логина и пароля) заключается в преобразовании

каждого символа строки в его Unicode, умножении на два и сохранении результата.

При дешифровании строка восстанавливается путем деления каждого числа на два и преобразования обратно в символ.

# Тестирование



* + 1. **Описание отчета о тестировании**

Данный отчёт предназначен для документирования проведенных тестов, их результатов и выводов.

# Цель тестирования

Цель тестирования заключается в проверке функциональности, производительности и удобства использования системы, а также в выявлении дефектов, которые могут повлиять на ёё работу.

# Методика тестирования

Тестирование проводилось с использованием следующих методов:

* + - * Ручное тестирование
      * Автоматизированное тестирование

# Проведенные тесты

В ходе тестирования были проведены следующие тесты:

1. Ручное тестирование.

Написаны и проведены следующие тест-кейсы и чек-листы: ТК1. Отработка авторизации.

Предварительные шаги:

Зарегистрироваться с логином: 12345 и паролем 12345. Шаги:

* 1. Запустить приложение.
  2. В окне регистрации, в поле «логин» ввести 12345, а в поле «пароль» 

12345.

* 1. Нажать кнопку «Войти».

Ожидаемый результат:

Открытие главного окна.

Фактический результат:

Открытие главного окна.

ТК2. Прохождение регистрации одно символьными логином и паролем.

Шаги:

1. Запустить приложение.
2. В окне регистрации, в поле «логин» ввести 1, а в поле «пароль» ввести 1.
3. Нажать на кнопку «Зарегистрироваться».

Ожидаемый результат:

Пользователь получит сообщение об ошибке. Фактический результат:

Пользователь получил сообщение об ошибке.

**Чек-лист для формы регистрации**

|  |  |  |
| --- | --- | --- |
| Описание | Пример | Результат |
| Логин не может быть меньше 3 символов | Логин: a | Появляется уведомление: "Логин должен быть от 3 символов". |
| Пароль не может быть меньше 4 символов | Пароль: 123 | Появляется уведомление: "Пароль должен быть от 4 символов". |
| Логин не может быть больше 8 символов | Логин: 123456789 | Не даёт ввести больше 8 символов. |
| Пароль не может быть больше 16 символов | Пароль: 12345678911111111 | Не даёт ввести больше 16 символов. |
| Поле логина и пароля пустые | Логин: Пароль: | Появляется уведомление: " Ошибка: Пустое поле логина или пароля!". |

|  |  |  |
| --- | --- | --- |
| При регистрации сохраняются данные (логин и пароль) | Логин: 12345, Пароль:  12345 | После нажатия на  "Зарегистрироваться" данные сохраняются, выводится сообщение "Логин и пароль сохранены". |
| При неверном логине или пароле появляется  ошибка | Логин: 123434, Пароль:  124341234 | После нажатия на "Войти", выводится сообщение " Неправильный логин или пароль". |

1. Автоматизированное тестирование

Для курсовой работы было составлено 7 unit-тестов. Тесты покрывают следующие функции приложения:

* **test\_encode\_data**: Проверяется, что функция encode\_data правильно кодирует строку. Мы проверяем, что результат кодирования соответствует ожиданиям (каждый символ строки умножается на 2).
* **test\_decode\_data**: Проверяется, что функция decode\_data правильно декодирует список чисел обратно в строку.
* **test\_save\_data**: Проверяется, что функция save\_data записывает данные в файл, и что в файле появляются сохраненные логин и пароль.
* **test\_check\_data**: Проверяется, что функция check\_data корректно проверяет правильность логина и пароля.
* **test\_create\_checkers\_board**: Проверяется, что функция создания доски возвращает доску правильного размера и с начальной расстановкой фигур.
* **test\_capture\_piece**: Проверяется, что функция capture\_piece корректно удаляет фигуру с доски после захвата.
* **test\_get\_possible\_moves**: Проверяется, что функция get\_possible\_moves возвращает правильные возможные ходы для фигуры.

Все unit-тесты прошли успешную проверку

# Выводы

На основе проведенных тестов сделаны следующие выводы:

* Программа успешно прошла все тесты и работает корректно.
* Рекомендации по дальнейшему развитию программы: добавление ограничения времени на ход пользователя, звукового сопровождения, таблицы лидеров, добавление краткой сводки правил.