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»**

# **Пояснительная записка**

**Тема:** Компьютерная логическая игра «Апит Содок»

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Р.02069337.22/3457-49 ПЗ-03

Листов 6

**Исполнитель**:

студент гр. ИСТбд-21

*Орешин Дмитрий Юрьевич*

« » 2023 г.

**2023**

**Введение**

Приложение «Апит-Содок» реализует функционал логической игры «Апит Содок»

В качестве подхода для разработки была выбрана каскадная модель разработки («Водопад»). Каскадная модель была выбрана из-за своей простоты, она позволяет наглядно представить объём работы и сроки выполнения, а также эффективно разбить проект на несколько подзадач.

Приложение «Апит-Содок» представляет собой игру Апит Содок на 64 клеточном поле (8 клеток в ширину и 8 в длину) для двоих игроков.

**1. Проектная часть**

**1.1 Постановка задачи на разработку приложения**

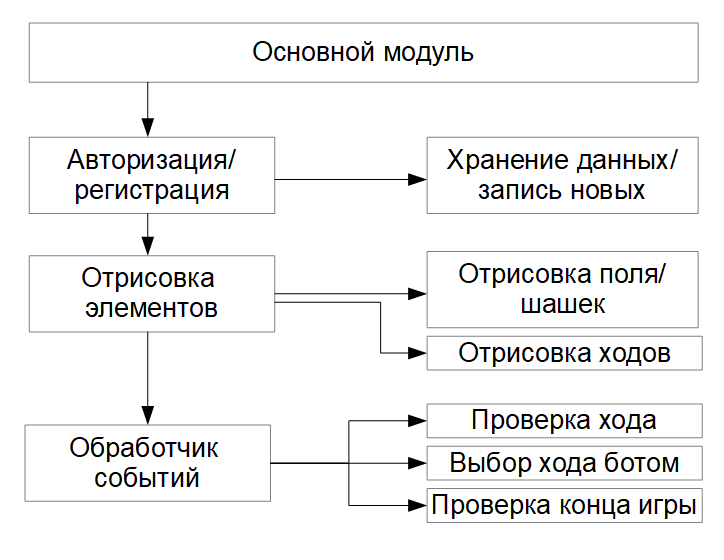
Определяется заданием на курсовую работу. Детализируется в разработанном техническом задании.

**1.2 Математические методы**

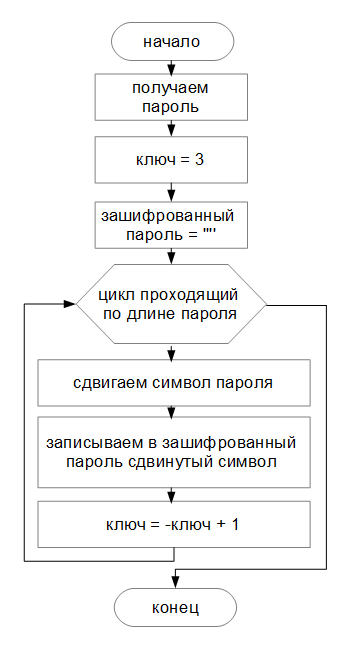
В качестве математической модели для представления и отрисовки поля был выбран двумерный массив, он позволяет легко записать положение всех шашек, выполнять все необходимые проверки и отрисовывать шашки, а также может быть легко изменен, что упрощает вывод хода пользователя. В массиве пустые клетки представлены числом 0, белые шашки 1, красные шашки 2.

**1.3 Архитектура и алгоритмы**

1.3.1. Архитектура

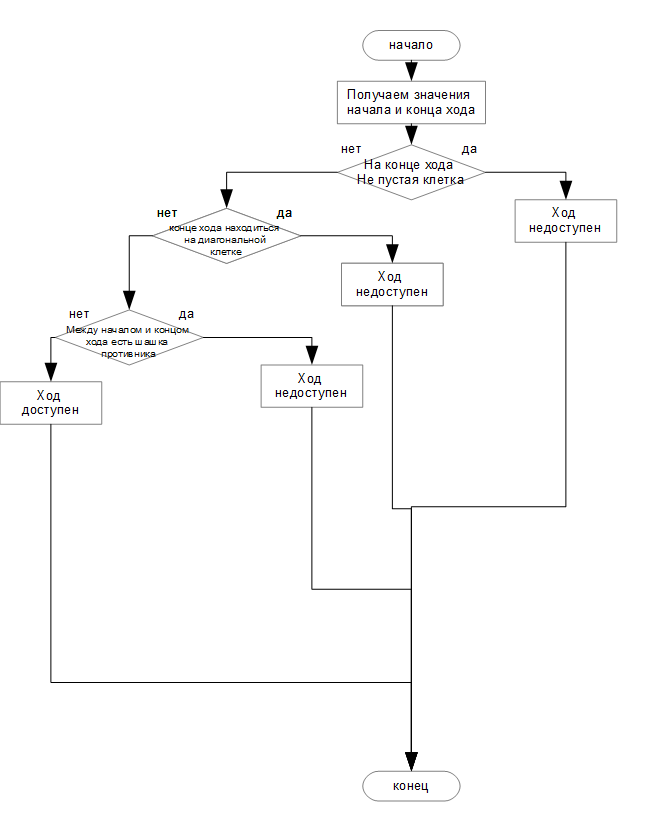


1.3.2.1. Алгоритм проверки регистрации



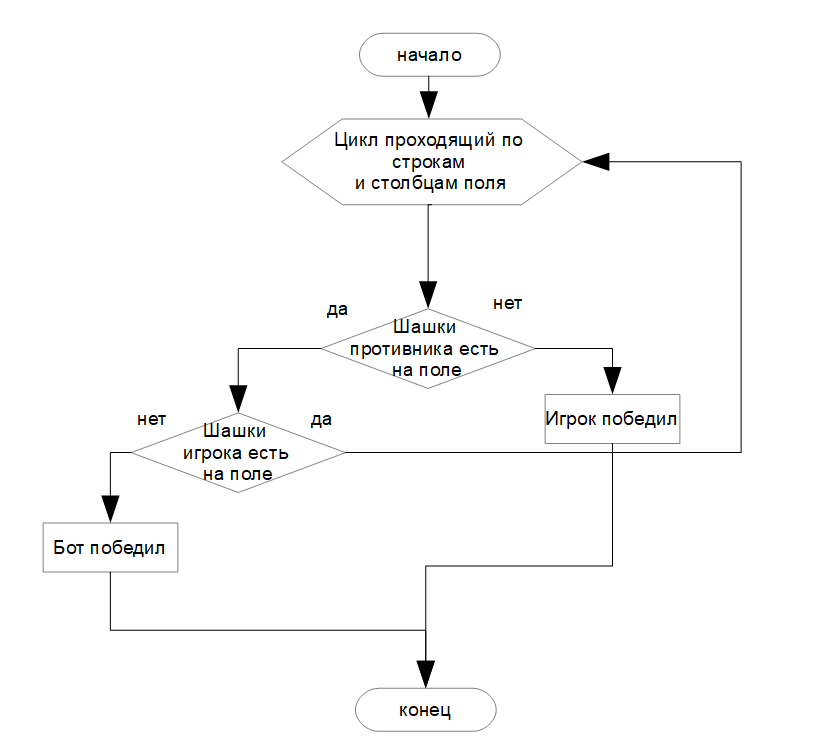
Данный алгоритм необходим для обеспечения безопасности данных пользователей. Алгоритм принимает на вход пароль пользователя, шифрует его с помощью ключа и возвращает зашифрованный пароль.

1.3.2.3. Алгоритм проверки доступности хода



Данный алгоритм необходим для контроля за соблюдением пользователем правил игры. Алгоритм проверяет возможен ли ход, который выбрал пользователь. Алгоритм получает на вход два индекса двумерного массива. Первый индекс указывает на шашку, которой пользователь хочет сходить, а второй на клетку, в которую он хочет сходить. Алгоритм проверяет возможность такого хода и, если он возможен двигает шашку пользователя, иначе ничего не делает и пользователь должен заново выбрать поле для хода.

1.3.2.4. Алгоритм проверки конца игры



Данный алгоритм необходим для того, чтобы определить победителя и корректно закончить игру. Алгоритм проверяет наличие шашек в случае их отсутствия запускает сценарий конца игры, иначе ничего не делает и игра продолжается.

**1.4 Тестирование**

Весь процесс тестирования проходил вручную, без привлечения специального ПО. На протяжении всего хода разработки, использовался метод белого ящика, так как в любом время имелся доступ ко всем компонентам программы. Всё тестирование выполнялось интуитивным методом, без подготовки специальных тестов.

На протяжении всего хода разработки, по мере добавления новых функций программы, использовалось системное тестирование новых функций, для устранения возникших в ходе написания ошибок. После положительных результатов тестирования функция считалась внедренной.

**2. Источники, использованные при разработке**

1. Шишкин, В.В. Разработка логических компьютерных игр с графическим интерфейсом в среде Python [Электронный ресурс] / В.В. Шишкин, Д.С. Афонин. – Ульяновск: УлГТУ, 2023. – 89 с. – Режим доступа: для всех пользователей. – URL: <http://lib.ulstu.ru/venec/disk/2023/112.pdf> (дата обращения: 22.10.23).

2. Руководство по программированию на Tkinter и Python [Электронный ресурс]. – Режим доступа: для всех пользователей. – URL: <https://metanit.com/python/tkinter/?ysclid=lq9y8691w1850842283> (дата обращения: 26.10.23