**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 логическая игра «Шашки Артамонова»

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Пояснительная записка

Р.02069337.21/1673-15 ПЗ-03

Листов 10

**Руководитель разработки**:

профессор каф. ИВК, к.т.н., доцент

*Шишкин Вадим Викторинович*

« » 2022 г.

**Исполнитель**:

студент гр. ИСТбд-22

Заикин Михаил Николаевич

« » 2022 г.

**2022**

**Введение**

Разрабатывается приложение «Шашки Артамонова». Данное приложение представляет из себя игру наподобие шашек с особыми правилами игры. В работе выбраны такие структуры данных, как массив и кортеж. Структура данных — массив была выбрана потому, что эта структура имеет множество преимуществ:

1. Массивы обеспечивают произвольный доступ к элементам. Это ускоряет доступ к элементам по положению;
2. Массивы хранят несколько данных похожих типов с одним и тем же именем;
3. В массиве данные организованны таким образом, что ими легко и удобно манипулировать.

Структура данных —кортеж был выбран, потому что он также обладает рядом преимуществ:

1. Кортежи выгодно использовать в случаях, когда нужно быть уверенным в целостности данных;
2. Кортеж не подвергается изменениям;
3. Он аналогичен списку, но занимает меньше памяти**.**

**1. Проектная часть**

**1.1 Постановка задачи на разработку приложения**

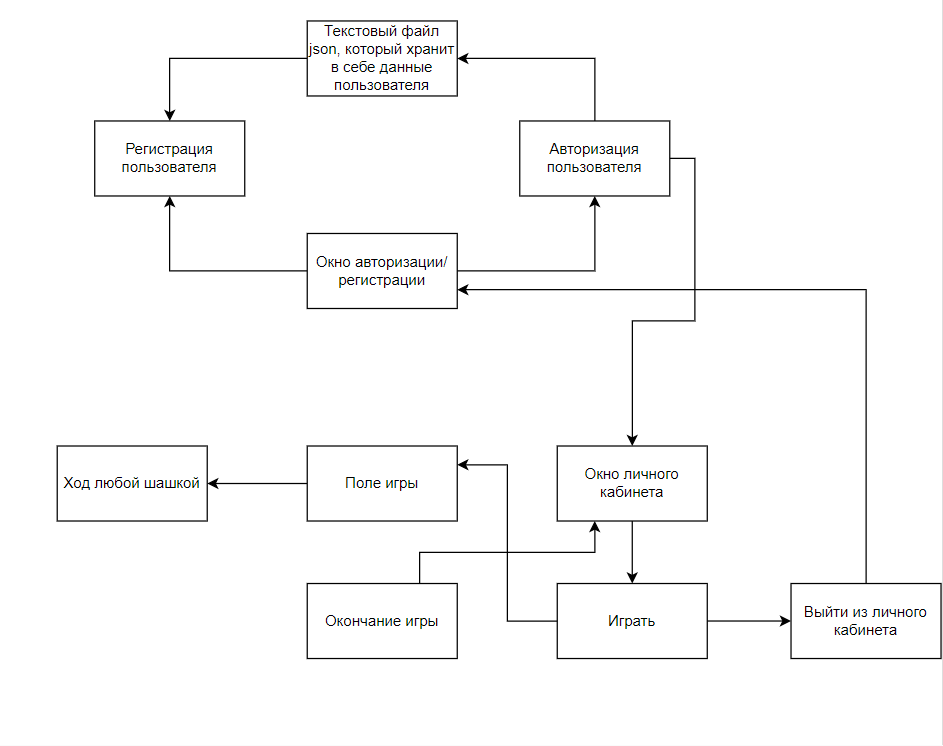
Определяется общей постановкой задачи в задании на курсовую работу.

**1.2 Математические методы**

Математические методы не применялись.

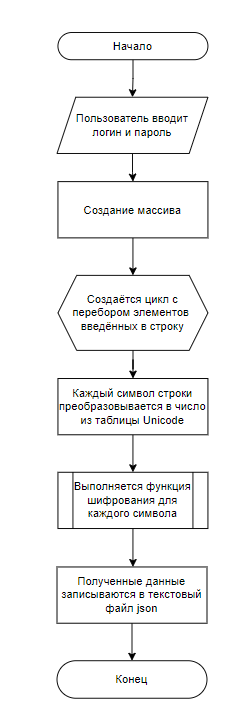
**1.3 Алгоритмы**

1.3.1. Структура приложения



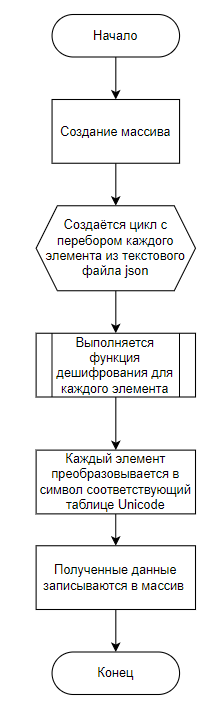
1.3.2 Алгоритм шифрования

Данный криптографический алгоритм предназначен для защиты логина и пароля пользователя. Каждый символ введённой строки сначала преобразовывается в число из таблицы Unicode, а потом шифруется с помощью дополнительной функции. Зашифрованные данные записываются в текстовый документ json.



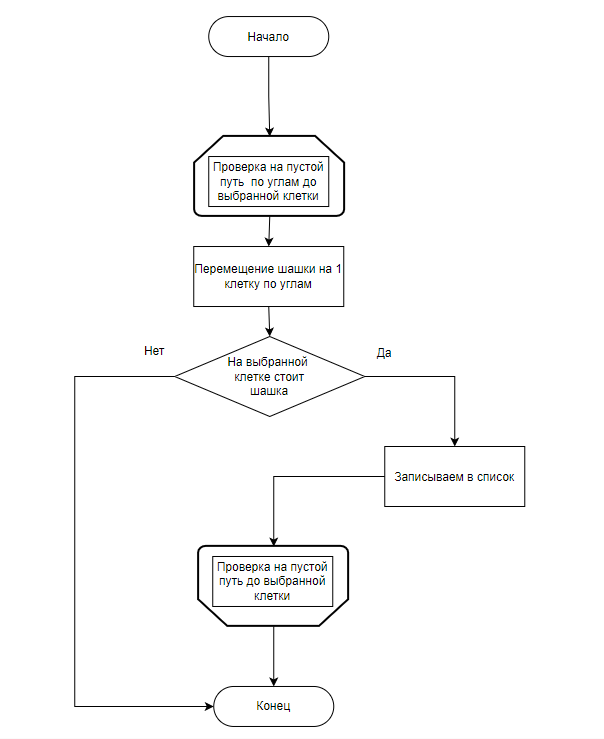
1.3.3 Алгоритм дешифрования

Данный алгоритм осуществляет расшифровку логина и пароля пользователя. Каждый элемент сначала дешифруется с помощью дополнительноой функции, а потом преобразовывается в символ из таблицы Unicode. После алгоритм сравнивает расшифрованные данные с введёнными данными пользователем.



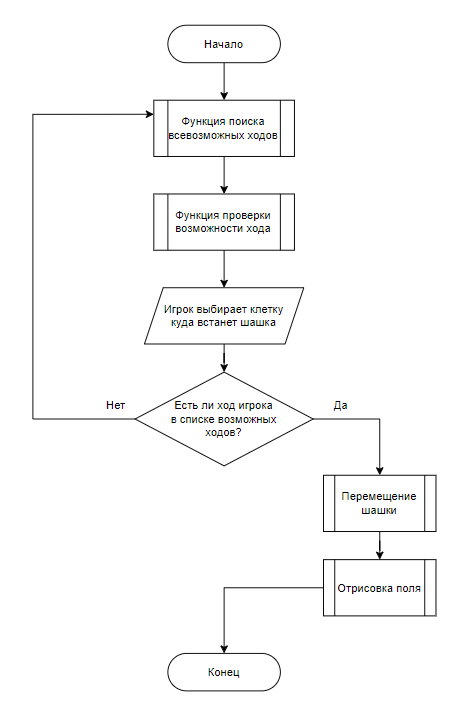
1.3.4 Алгоритм проверки возможности хода

Данный алгоритм проверяет возможность сходить шашкой в соответствии с правилами игры. Проверяются клетки от выбранной до текущей. Если они пустые, то ход возможен.



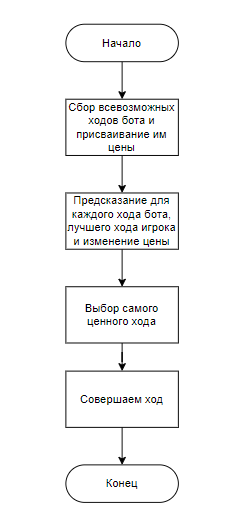
1.3.5 Алгоритм хода игрока

Данный алгоритм совершает ход игрока. Для начала идёт выявление всевозможных ходов и функция на возможность хода. После выбора хода игрока, шашка переририсовывается.



1.3.6 Алгоритм хода компьютера

Данный алгоритм осуществляет ход противника(компьютера). Данный алгоритм совершает ход случайной шашкой, соответсвуя правилам игры.



1.3.7 Алгоритм регистрации

Данный алгоритм осуществляет регистрацию пользователя, зашифровывая данные с помощью алгоритма шифрования и записывая данные в текстовый файл json

1.3.8 Алгоритм авторизации

Алгоритм осуществляет авторизацию пользователя. Пользователь вводит логин и пароль, а затем алгоритм проверяет зарегистрирован ли такой пользователь.

1.3.9 Алгоритм загрузки изображений

Алгоритм осуществляет загрузку картинок пешек и дамок.

1.3.10 Алгоритм создания окна приложений

Данный алгоритм создаёт основное окно приложения.

1.3.11 Алгоритм новой игры

Алгоритм реализует запуск новой игры.

1.3.12 Алгоритм рендеринга поля

Алгоритм используя входные данные, отрисовывает игровое поле с шашками.

1.3.13 Алгоритм перемещения мышки

Данный алгоритм позволяет мышке двигаться по полю и подсвечивать клетку зелёной рамкой.

1.3.14 Алгоритм выбора клетки

Алгоритм позволяет выбрать любую шашку на поле и получает координаты этой шашки подсвечивая поле выбранной шашки красным контуром.

1.3.15 Алгоритм Возможных ходов

Алгоритм проверяет всевозможные ходы и записывает их в список.

1.3.16 Алгоритм счётчика

Данный алгоритм считывает ситуацию на поле, записывая сколько съел игрок и бот, подсчитывая эти данные.

1.3.17 Алгоритм конца игры

Алгоритм проверяет ситуацию на поле и в соответствии с правилами игры выводит сообщение с данными победителя.

1.3.18 Алгоритм просмотра хода выбранной шашки

Данный Алгоритм проверяет правильность хода для выбранной шашки в соответствии с правилами игры.

1.3.19 Алгоритм передачи хода

Данный алгоритм реализует передачу хода от игрока к боту

1.3.20 Алгоритм проверки передачи хода

Данный алгоритм проверяет передачу хода ссылаясь на правила игры.

**1.4 Тестирование**

Системное тестирование.

При системном тестировании наша задача уже состоит в том, чтобы убедиться в корректности работы в целом всей системы. На данном этапе проводятся следующие тесты:

* Функциональное тестирование;
* Тестирование удобства пользования;
* Тестирование на отказ и восстановление.

Тесты:

Симулировать внезапный отказ электричества на компьютере (обесточить компьютер).

Симулировать ситуацию наличия в системе неверных данных (специальный тестовый набор или база данных).

**2. Источники, использованные при разработке**

1. Wikipedia [Электронный ресурс]: Шашки Артамонова (дата обращения 4.11.2022);
2. Wikipedia [Электронный ресурс]: RSA (дата обращения 19.11.2022);
3. Постолит А.В. Основы искусственного интеллекта в примерах на Python. Самоучитель. [Электронный ресурс]: книга / СПб.: БХВ-Петербург, 2021. -448 с.: ил. -(Самоучитель) (дата обращения 27.10.2022).