**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 «Фризские шашки – Поддавки»

Пояснительная записка

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Р.02069337. 22/384-17 ПЗ-01

Листов 8

**Руководитель разработки**:

доцент каф. ИВК, к.т.н., доцент

*Шишкин Вадим Викторинович*

« » 2022 г.

**Исполнитель**:

студент гр. ИСТбд-22

*Игнатьев Дмитрий Сергегевич*

« » 2022 г.

**2022**

**Введение**

Приложение “ Фризские шашки - поддавки ”. Фризские шашки , которые также называют фрисландскими – вариант игры в "Шашки" на 100–клеточной доске, являющаяся предшественником международных шашек с некоторыми отличиями в правилах боя. В приложении представлена модификация фризских шашек "Поддавки", в отличии от стандартной версии для победы нужно лишить себя возможности ходить путём умышленной отдачи всех своих фигур сопернику. Приложение включает в себя регистрацию и авторизацию, а также возможность играть в фризские шашки-поддавки против компьютера. В работе выбрана структура данных – массив, так как он многофункционален и имеет ряд преимуществ:

1. Для формирование доски используется Массив.
2. Массивы хранят одноименный ряд данных похожих типов с одинаковым именем.
3. Массивы обеспечивают произвольный доступ к элементам, что существенно ускоряет доступ к элементам по положению.

**1. Технические характеристики**

**1.1 Постановка задачи на разработку приложения**

Определяется общей постановкой задачи в задании на курсовую работу.

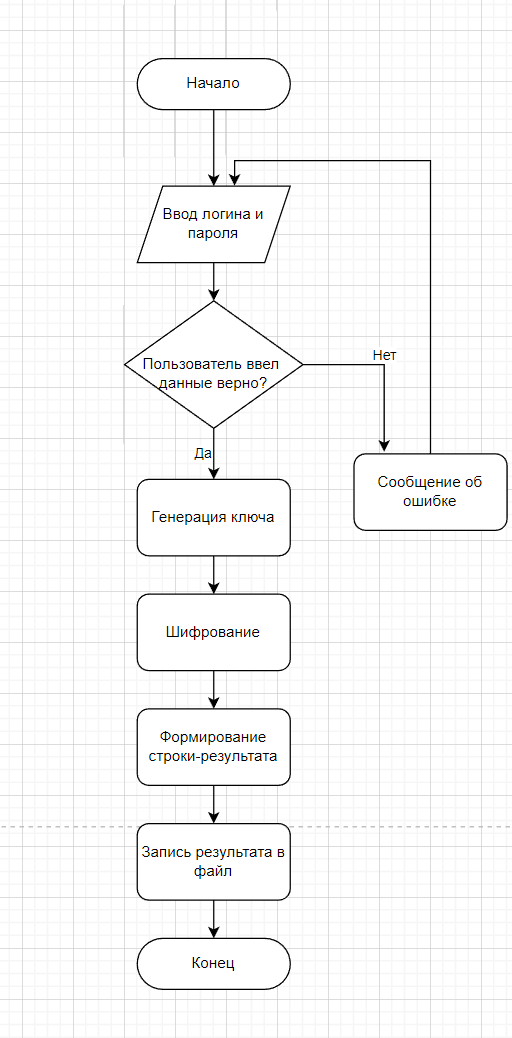
**1.2 Математические методы**

Математический аппарат не используется.

**1.3 Алгоритмы**

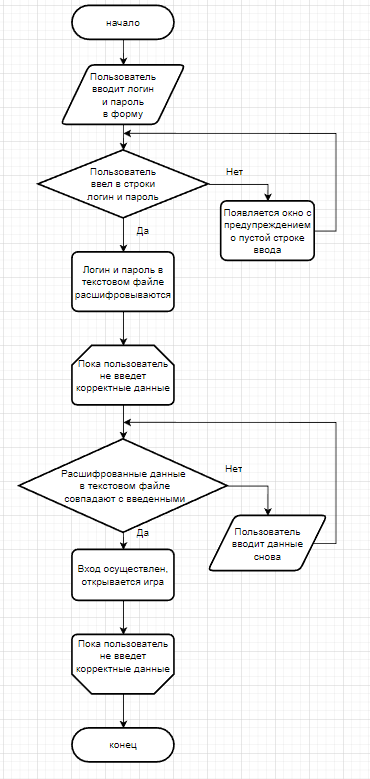
1.3.1 Алгоритм Шифрование

Алгоритм выполняет регистрацию пользователя и шифрование введёных данных пользователя в текстовый файл. Пользователь вводит логин и пароль в форму, после чего идет проверка на пустые строки в случае, если пользователь ошибочно ввел данные - появится окно с предупреждением, если же пользователь ввел логин и пароль, то данные шифруются и записываются в текстовый файл.



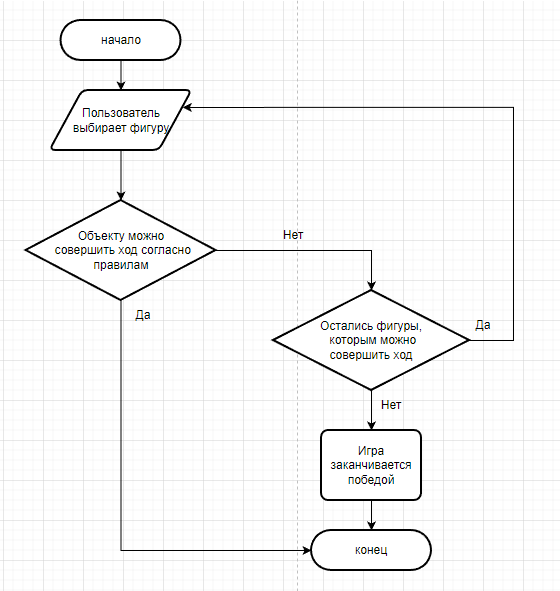
1.3.2 Алгоритм Дешифрование

Алгоритм выполняет авторизацию пользователя и дешифрацию данных из текстового файла. Пользователь вводит логин и пароль в форму, если он задал пароль в неверном формате или ввёл несуществующие данные – выводится ошибка . если же пользователь ввел логин и пароль, то начинается проверка на корректность, данные в текстовом файле расшифровываются и сверяются с данными, которые ввёл пользователь в случае, если данные совпали, пользователь может войти в личный кабинет и поиграть в фризские шашки.



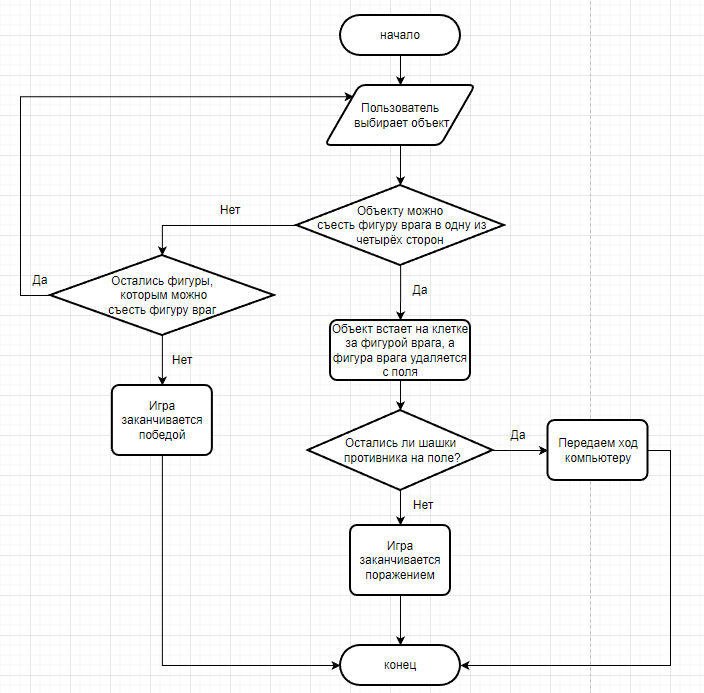
1.3.3 Алгоритм Проверка доступности хода

Данный алгоритм выполняет проверку шашки на доступность хода, в любую из возможных сторон. Алгоритм проверяет у выбранной шашки возможные варианты хода, если шашке на пути ничего не преграждает, то шашка может сходить в одну из четырёх сторон.



1.3.4 Алгоритм Процесс хода

Данный алгоритм обрабатывает процесс хода игрока. Алгоритм проверяет может ли шашка или дамка забрать противника, если да, то проверяет в какую из сторон можно забрать, после чего производится процесс съедения, удаляется шашка противника, место, где стояла шашка игрока также освобождается, шашка игрока становится на поле за врагом.



1.3.5 Алгоритм Проверка на возможность забрать шашку врага

Данный алгоритм проверяет все возможные ситуации на доске, когда шашка или дамка может забрать шашку или дамку врага, если такая ситуация была найдена, то шашке можно съесть в эту сторону.

1.3.6 Алгоритм подсвечивания возможных ходов

Алгоритм используя данные, полученные из алгоритма возможности хода, подсвечивает зеленым все возможные ходы.

**1.4 Тестирование**

Уровни тестирования:

1. Компонентное/модульное тестирование.

Этот вид тестирования выполняется на самой ранней стадии разработки программы — во время написания кода. На данном этапе проводятся следующие тесты:

* Функциональное тестирование.

Тесты:

Для окон авторизации и личного кабинета: проверка корректной работы кнопок.

Для игрового поля: проверка корректного отображения поля и шашек, проверка игры по правилам.

* Тестирование безопасности.

Тесты проводятся для окна авторизации, конкретно тестируются алгоритмы регистрации, авторизации, шифрования и дешифрования.

1. Интеграционное тестирование.

Интеграционное тестирование необходимо для того, чтобы тестировать взаимосвязь между модулями. Проводится тестирование взаимодействия 3 окон: авторизации, личного кабинета и игрового поля.

1. 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]: Фризские шашки – URL: https://ru.wikipedia.org/wiki/Фризские\_шашки (дата обращения: 7.12.2022)
2. Wikipedia [Электронный ресурс]: https://ru.wikipedia.org/wiki/RSA (дата обращения: 9.12.2022)

**3. Приложения**

В приложении 1 приводится техническое задание, в приложении 2 приводится руководство программиста.