Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**Отчёт по лабораторной работе**

по дисциплине «Технология разработки качественного ПО»

Приложение «игра в ассоциации»

Выполнил: студент гр.

3530904/70101 Мосин А.В.

Руководитель: Смирнов Н.

1. Цель работы

Необходимо реализовать игру на java. Ограничений нет. Можно использовать любые дополнительные библиотеки.

Требования: Словесная игра - необходимо угадать слово по набору ассоциаций (слова или фразы из ассоциативного словаря - английский или русский). Статья ассоциативного словаря состоит из СЛОВА - <ассоциаиция1>X, <ассоциация2>Y..., где X и Y - частотность ассоциации (или её вес). Разработать правила игры, предусмотреть несколько уровней сложности. Персистенция результатов игроков в файл

1. Обзор предметной области

Приложение должно быть написано на Java. Ограничений на используемые библиотеки нету.

Приложение запускается из IDE

* должна быть реализована возможность игры одного/нескольких человек.
* выбор различных уровней сложности
* возможность сохранить/загрузить прогресс
* вывод статистики/результатов

Пользователь может:

* начинать новую игру
* завершать текущую игру
* осуществлять ход (ввод слова, пропуск слова, получение дополнительной ассоциации)
* задавать сложность перед началом игры
* при ошибочном вводе – переход к следующему игроку
* неограниченное количество игроков

1. Сравнение аналогичных программ

Сравним данную реализацию с игрой крестики-нолики

<https://javarush.ru/groups/posts/1228-mnogopoljhzovateljhskaja-konsoljhnaja-igra-na-java>

|  |  |  |
| --- | --- | --- |
| **Реализация** | **Крестики-нолики** | **Ассоциации** |
| Сохранение данных | БД MySQL | Файл |
| Количество участников | 2 | Любое количество |
| Вывод статистики | Вывод в консоль через println | Вывод в консоль через println |
| Сериализация | Нет | Стандартные средства Serializable |

Считаю, что для таких простых приложений нет смысла использовать БД и hibernate, т.к. данных небольшое количество. Самый простой вариант – сохранять в файл и шифровать при необходимости

1. Описание работы приложения
   1. начало игры. Начинаем новую игру или продолжаем сохраненную

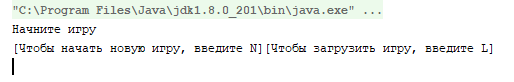


Рисунок 1. Начало игры

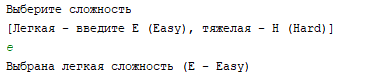
* 1. выбор уровня сложности

Рисунок 2. Выбор сложности

* 1. ввод имен игроков

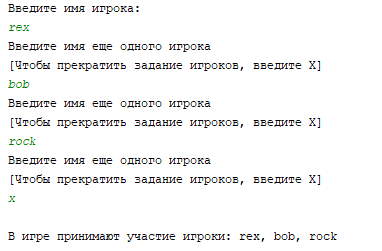


Рисунок 3. Ввод игроков

* 1. ввод ответа



Рисунок 4. Игровой процесс

* 1. сохранение игры и результат

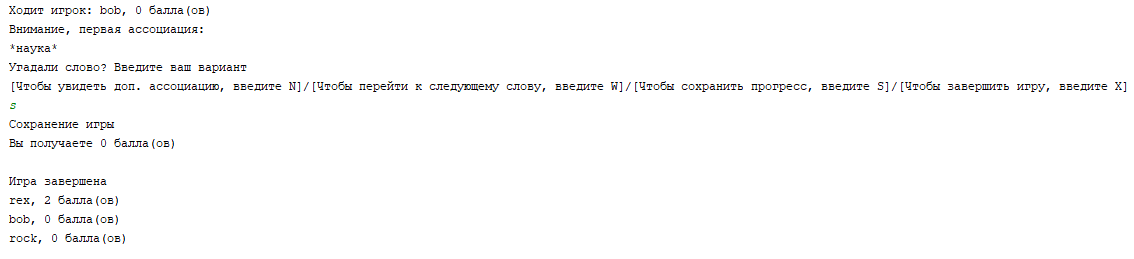


Рисунок 5. Сохранение и вывод результат

1. Обоснование дизайна и диаграмма

Использованы были стандартные возможности встроенной библиотеки Java без излишеств.

Чтение словаря выполнялось с помощью try с ресурсами. Все необходимые исключения были обработаны.

Интерактив с пользователем выполнен через System.out-println и ввод клавиш от пользователя. Значения клавиш для команд были вынесены в константы.

Персистенция и сохранение/загрузка прогресса реализованы через стандартный интерфейс Serializable.

Для сохранения порядка слов в словаре использовал TreeMap

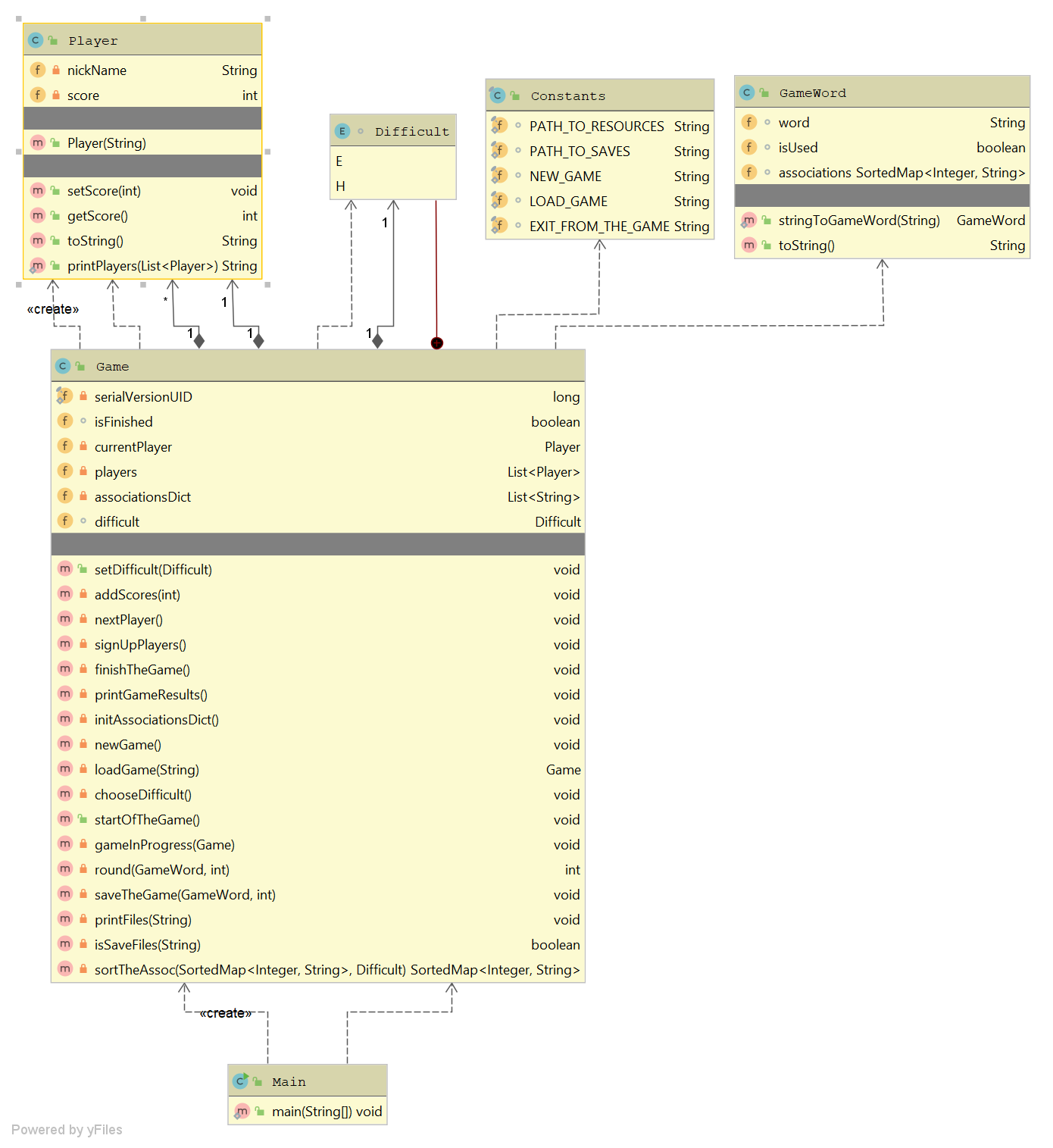


Рисунок 6. UML диаграмма классов

1. Выводы

В ходе выполнения работы были получены навыки в написании простейших консольных Java приложений. Получены навыки работы с сериализацией объектов, работой с пользователем, обработкой ошибок.

Для улучшения стабильности программы необходимо наличие юнит-тестов, которые написаны не были.

Так же необходимо дополнительная декомпозиция классов и методов приложения. Механизм работы с пользователем следует улучшить и добиться единообразия