

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

## Курсовая работа

Дисциплина: Алгоритмы и структуры данных

Тема: Решатель игры “Wordle”

Выполнил студент гр. 3530901/00002 \_\_\_\_\_ Разин М. А.  
(подпись)

Принял старший преподаватель \_\_\_\_\_ Степанов Д. С.  
(подпись)

“ \_\_\_\_ ” \_\_\_\_\_ 2022 г.

Санкт-Петербург  
2022

## ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

студенту группы 3530901/00002 Разину Макару Александровичу

1. **Тема проекта (работы):** Реализация игры “Wordle” и её решателя
2. **Срок сдачи законченной работы** 27.05.2022
3. **Исходные данные к работе:** IDE: Android Studio Chipmunk 2021.2.1
4. **Содержание пояснительной записки:** (перечень подлежащих разработке вопросов): введение, основная часть (текст программы, описание программы, теоретическая часть, испытания программы), заключение, список использованных источников

Дата получения задания: « 8 » февраля 2022 г.

Руководитель \_\_\_\_\_ Степанов Д. С.  
(подпись)

Задание принял к исполнению \_\_\_\_\_ Разин М. А.  
(подпись)

\_\_\_\_\_  
(дата)

## Исходные данные к работе

Вариант – Реализация игры “Wordle” и её решателя. Проверить корректность выполняемой работы тестами.

Первым мы ввели слово Ребус. Буква **С** есть в загаданном слове, но стоит в другом месте.

РЕБУС

Затем ввели слово Сосна. Буква **С** и **О** есть в загаданном слове и стоят на правильных местах.

СОСНА

Обратите внимание, что если в введенном слове две одинаковых буквы, а в загаданном слове только одна такая буква, то выделяется только одна буква.

Если слово угадано правильно, то все буквы будут выделены!

ГОСТЬ

Если буквы нет в загаданном слове, то она выделяется серым. Также важно знать, что в загаданном слове могут быть одинаковые буквы. По правилам русских кроссвордов буква Ё в словах заменена на Е!

ПИРАТ

Рис. 1. – Правила игры в “Wordle”.

## Текст программы

<https://github.com/MonkeyProger/KtWorlde>

## Описание программы

В качестве реализации решателя игры “Wordle” был подготовлен алгоритм нахождения 5 лучших слов, позволяющих максимально сократить количество попыток угадать слово, на определенном шаге. Слова в этом списке упорядочены по убыванию значений информационной энтропии, которую они представляют.

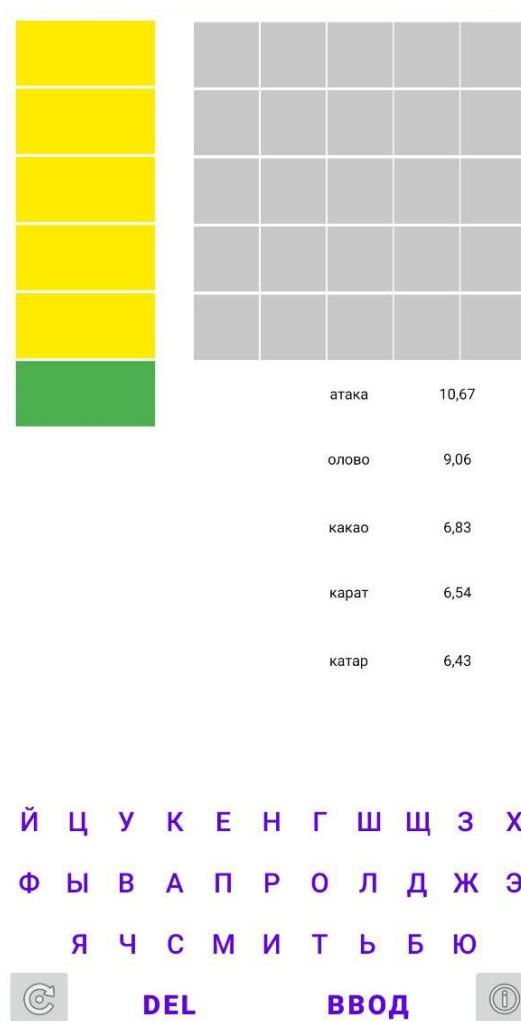


Рис. 2. – Запущенная программа с полученным списком “подсказок” до момента введения слова (т. е. для всего словаря).

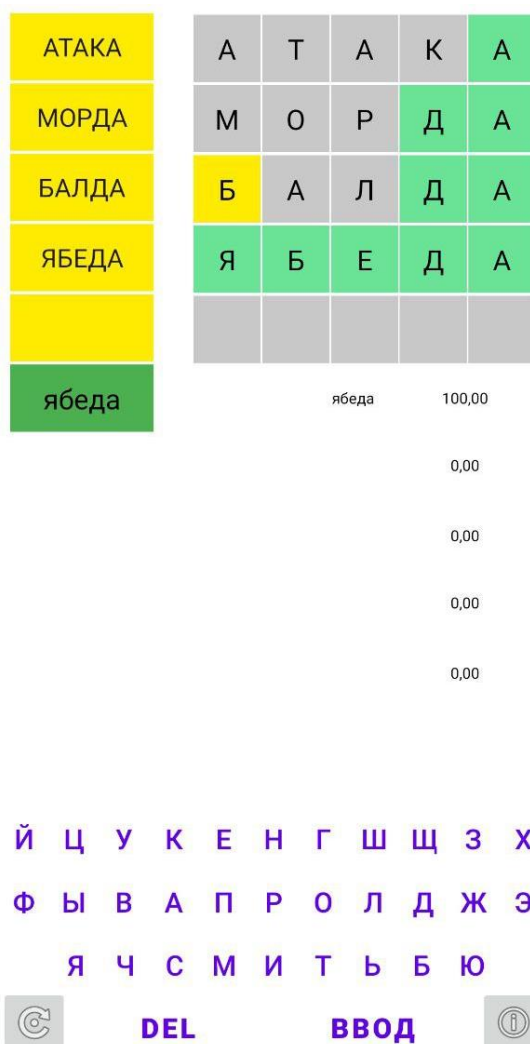


Рис. 3. – Получение загаданного слова с отраженными попытками/шагами его угадать.

## Теоретическая часть

В теории информации энтропия — это средняя скорость генерирования значений некоторым случайным источником данных. Величина информационной энтропии, связанная с определенным значением данных, вычисляется по формуле:

$$E[I] = \sum_x p(x) \cdot \log_2 \left( \frac{1}{p(x)} \right), \text{ где } p(x) \text{ — вероятность состояния } x \text{ системы.}$$

Когда источник данных генерирует значение, имеющее низкую вероятность (т.е. когда происходит маловероятное, неожиданное событие), с ним связана большая информация, чем с более вероятным событием. Количество информации, выражаемое событием, связанным с появлением определенного значения данных,

можно рассматривать как случайную переменную, математическое ожидание которой равно информационной энтропии. Чем ближе распределение к равномерному, тем выше энтропия.

Таким образом, информационную энтропию можно рассматривать как меру неупорядоченности или неопределенности состояния некоторой системы, описываемой данными.

## **Класс WordAnalyzer**

Класс, в котором реализованы функции сравнения слов, их фильтрации, нахождения энтропии, составления списка 5 слов, имеющих наибольшую энтропию.

Содержит в себе значение ответа, словари (возможных для введения слов и возможных ответов), список подходящих слов на текущем шаге.

### ***analyzeWord(word: String)***

Функция сравнивает слово с ответом и составляет комбинацию, которая отражает положение и наличие букв во введенном слове по отношению к ответу.

### ***analyzeEntropy(word: String)***

Функция возвращает топ 5 слов, дающих наибольшую энтропию после введения слова.

### ***analyzeFullEntropy()***

Функция возвращает топ 5 слов, дающих наибольшую энтропию до введения слова.

### ***getE(word: String)***

Функция получает энтропию для слова по списку всех возможных слов из словаря. Для нахождения энтропии для слова необходимо проверить каждую возможную комбинацию из игры.

### ***buildMatches(word: String, comb: IntArray)***

Функция для фильтрации списка подходящих слов по конкретной комбинации к слову.

***buildPos(word: String, comb: IntArray)***

Функция нахождения числа подошедших слов к догадке word с комбинацией comb.

***isValid(word, String, comb, IntArray, i: String)***

Функция реализует сравнение текущего слова i с введенным предположенным словом word. Возвращает true, если слово подходит.

## **Испытания программы**

Для программы написаны тесты (ExampleUnitTest, AbstractFunctionTest), проверяющие корректность выполнения функций, время их выполнения, а также тесты проходящие весь цикл угадывания слова с отражением соответствующих данных: первоначальное введенное слово, ответ, количество итераций потребовавшихся алгоритму угадать слово (дополнительно найдены средние значения итераций для определенного слова).

***analyzeWordTest()***

Проверка корректности вывода комбинации заданного слова при определенном ключевом слове.

***validationTest()***

Проверка того, подходит ли заданное слово введенному при определенной комбинации.

***buildPosTimeTest(word: String)***

Тест времени выполнения функции buildPos() для выбранного слова word.

***getEntropyTimeTest(word: String)***

Тест времени выполнения функции getE() для выбранного слова word.

***analyzeFullEntrTimeTest()***

Тест времени выполнения функции `analyzeFullEntropy()`, т.е. получения списка лучших слов до момента введения первого слова.

***iterationsOnGuessTest(key: String, word: String)***

Получения числа итераций до получения ответа `key` по заданному слову `word`.

В классе `ExampleUnitTest` тест `iterationsOnGuessTest()` применен к выбранному слову-эталону “олово” и к случайно выбранному второму слову для сравнения (рис. 4). Также получены средние значения числа итерации соответственно к слову “олово” и к случайно выбранному слову (рис. 5).

```
Количество итераций для ответа 'ведро' при догадке 'олово':2  
Количество итераций для ответа 'ведро' при догадке 'комок':3
```

Рис. 4. – Результат выполнения `iterationsOnGuessTest()`.

```
Количество итераций в среднем для догадки 'олово':3  
Количество итераций в среднем для случайно выбранной догадки 'ройка':2
```

Рис. 5. – Результат выполнения `iterationsOnGuessTest()` для получения среднего числа итераций.



## **Заключение**

В результате выполнения курсового проекта мною были изучены алгоритмы применения информационной энтропии и других статистических критериев в различных задачах анализа данных. Была разработана модель игры “Wordle” и алгоритм получения наиболее “информативных” слов-подсказок для определенного шага в игре. Проведены тесты функций и времени их выполнения. Также составлены тесты получения информации о количествах итераций, понадобившихся для победы в игре для слова-эталона и случайно выбранного слова.

## Список использованных источников

- [https://ru.wikipedia.org/wiki/Информационная\\_Энтропия](https://ru.wikipedia.org/wiki/Информационная_Энтропия)  
- Википедия: Информационная энтропия.
- <https://wordle.belousov.one/>  
- Пример игры и правила к ней.
- <https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8>  
- Примеры использования энтропии в дереве решений.
- <https://www.youtube.com/watch?v=v68zYyaEmEA>  
- Solving Wordle using information theory