

## Задание №1

Будем реализовывать ‘text-analyzer’. Приложение запрашивает у пользователя путь к файлу и тип операции которую будет выполнять, после завершения операции выведется результат и пользователь снова будет опрошен.

### Операция №1 Имя = **frequency**

Вывести два наиболее встречаемых слова, слова отсортировать их в алфавитном порядку по убыванию, и вывести сколько раз слово встречается в тексте.

good -> 23

allow -> 2

### Операция №2 Имя = **length**

Найти первые три самых длинных слова, отсортировать их по длине по убыванию.

battle -> 6

map -> 3

a – 1

### Операция №3 Имя = **duplicates**

Найти первых три слова у которых есть дубликаты и распечатать их в обратном порядке в верхнем регистре отсортированные по длине по возрастанию (e.g. map -> ram)

PAM

WOLLA

STNEMUGRA

## Задание №2

Переопределить механизм генерации хеш-кода для строчных ключей (использовать два разных метода генерации: в качестве хеш-кода использовать длину строки и сумму первых четырех символов).

Для обоих случаев заполнить таблицу парами артикул (строка) – значение (объект товара).

Исследовать порядок вывода элементов встроенным итератором для обоих случаев генерации хеш-кода при использовании HashMap и LinkedHashMap.

## Задание №3

Выполнить задание из OOP programming in Java, сохраняя объекты приложения в одном или нескольких файлах с применением механизма сериализации. Объекты могут содержать поля, помеченные как static, а также transient. Для изменения информации и извлечения информации в файле создать специальный класс-коннектор с необходимыми для выполнения этих задач методами.

