

## Лабораторная работа №2

### Отображения, трейты и функции высшего порядка

#### Задание

Написать скрипт, принимающий в качестве первого параметра путь к текстовому файлу (plain text). Остальные параметры указаны в вариантах заданий.

В соответствии с вариантом задания скрипт должен вывести в стандартный поток результаты обработки, а в случае возникновения ошибки вывести соответствующее сообщение в поток ошибок (таким образом, скрипт можно будет использовать в unix/linux pipeline).

Конкретная реализация механизма чтения из файла остаётся на ваше усмотрение, однако, будьте готовы ответить на вопрос: “как нужно модифицировать скрипт, если размер файла значительно - до гигабайтов - возрастёт?”

Словом в тексте считаем любую последовательность непробельных (отображаемых) символов, которая ограничена пробельными символами либо началом/концом строки.

#### Варианты

1. Вернуть список слов из текста, начинающихся на указанную букву (задается как параметр). Каждое слово - с новой строки в алфавитном порядке.
2. Вернуть список слов из текста, заканчивающихся на указанную букву (задается как параметр). Каждое слово - с новой строки в алфавитном порядке.
3. Вернуть коэффициент заполнения для каждой строки текста (отношение непробельных символов к пробельным).
4. Вернуть среднюю длину слова для каждой строки текста.
5. Вернуть максимальную длину слова для каждой строки текста.
6. Вернуть количество в тексте для каждого непробельного (отображаемого) символа.
7. Вернуть N (задается как параметр) самых длинных слов в тексте (при равенстве длин - в алфавитном порядке).

8. Вернуть N (задается как параметр) самых коротких слов в тексте (но длиннее 1 символа) (при равенстве длин - в алфавитном порядке).
9. Вернуть количество уникальных слов в тексте (одно число) длиннее N (задается как параметр) символов.
10. Вернуть номера строк (через пробел), содержащих более одного N (задается как параметр) пробелов подряд.