

Домашняя работа 4. Grep

07.02.2022

Дедлайн: 10:00 21.02.2022

Баллов: 10

В тех же командах, что и раньше, реализовать в своём Command-Line Interface из предыдущих домашних заданий поддержку встроенной команды `grep`.

Требуется поддержка:

- регулярных выражений в запросе;
- ключа `-w` — поиск только слова целиком;
 - формально `grep -w` ищет подстроки, ограниченные «non-word constituent character», а именно не буквы, цифры или символ подчёркивания; однако что такое буквы — лучше спросить у вашей стандартной библиотеки, потому что бывает Unicode и его классы символов: впрочем, ваша реализация может отличаться от настоящего поведения `grep`;
- ключа `-i` — регистронезависимый (case-insensitive) поиск;
- ключа `-A` — следующее за `-A` число говорит, сколько строк после совпадения надо распечатать: например, `-A 0` печатает только строку, на которой найдено совпадение, а `-A 10` — ещё и 10 строк ниже;
 - подумайте, что будет, если области печати пересекаются.

Поскольку для `grep` требуется нетривиальный разбор ключей, его необходимо реализовать с использованием одной из библиотек для разбора аргументов командной строки (например, Apache Commons CLI). Выбрать библиотеку необходимо самостоятельно, и, что самое главное, *описать в README.md, из чего выбирали и почему выбрали именно ту, что выбрали.*

Примеры:

```
> grep 'Минимальный' README.md
Минимальный синтаксис grep
```

```
> grep 'Минимальный$' README.md
```

```
> grep '^Минимальный' README.md
Минимальный синтаксис grep
```

```
> grep -i 'минимальный' README.md
Минимальный синтаксис grep
```

```
> grep -w 'Минимал' README.md
```

```
> grep -A 1 'II' README.md
```

Нужно также:

- обновить архитектурную документацию и заодно привести диаграмму к синтаксису UML, если она ему не соответствует;
- написать юнит-тесты на новую функциональность;
- написать комментарии.

Сдавать, как обычно, создав новую ветку для `grep` на базе предыдущей и сделав новый пул-реквест.