

# Проектирование программного обеспечения (практика)

Практика 1: задача про CLI

Юрий Литвинов  
yurii.litvinov@gmail.com

07.02.2022

# Задача про CLI

В командах по два человека реализовать простой интерпретатор командной строки, поддерживающий команды:

- ▶ **cat [FILE]** — вывести на экран содержимое файла
- ▶ **echo** — вывести на экран свой аргумент (или аргументы)
- ▶ **wc [FILE]** — вывести количество строк, слов и байт в файле
- ▶ **pwd** — распечатать текущую директорию
- ▶ **exit** — выйти из интерпретатора

# Задача про CLI (продолжение)

- ▶ Должны поддерживаться одинарные и двойные кавычки (full and weak quoting)
- ▶ Окружение (команды вида “имя=значение”), оператор \$
- ▶ Вызов внешней программы
  - ▶ если введено что-то, чего интерпретатор не знает
- ▶ Пайплайны (оператор “|”)

# Примеры

```
>echo "Hello, world!"  
Hello, world!
```

```
> FILE=example.txt  
> cat $FILE  
Some example text
```

```
> cat example.txt | wc  
1 3 18
```

```
> echo 123 | wc  
1 1 3
```

```
> x=ex  
> y=it  
> $x$y
```

# Что ожидается в качестве решения

- ▶ Архитектурная документация, как умеете
  - ▶ Структурная диаграмма (классов, компонентов, квадратиков со стрелочками)
  - ▶ Словесное описание работы системы
  - ▶ Достаточно подробно, чтобы не требовалось принимать важные решения при кодировании
    - ▶ Не должно быть «ну тут мы парсим строку»
- ▶ Реализовывать проект пока не нужно

# Что делать дома

- ▶ Завести для этого курса репозиторий
- ▶ Одному из членов команды выложить решение в виде .md или .pdf-файла в отдельную ветку
  - ▶ Обязательно укажите, с кем в команде вы делали
- ▶ Сделать пуллреквест к себе в основную ветку
- ▶ Ссылку на пуллреквест приложить в качестве решения в LMS-системе CSC
- ▶ Смерджить пуллреквест, когда задача зачтена
- ▶ **Дедлайн: 10:00 14.02.2022**

# Что делать сейчас

## Первые фазы жизненного цикла

- ▶ Разбиться на команды по два человека
  - ▶ Если очень хочется, то три тоже ок
- ▶ Выполнить анализ и определить подходы к решению
- ▶ Выявить подводные камни и способы их преодоления
- ▶ Декомпонировать задачу на подсистемы, классы и методы
- ▶ Нарисовать первое приближение структурной диаграммы
- ▶ Быть готовыми в конце пары рассказать предлагаемое решение
- ▶ Дома это надо будет уточнить, расширить и оформить

# Соображения

- ▶ Проектирование сверху вниз
  - ▶ Определитесь с общей структурой системы
  - ▶ Определитесь с компонентами, их ответственностью и связями между ними
  - ▶ Только после этого переходите к проектированию компонентов
    - ▶ По такой же схеме
  - ▶ Возможно, придётся возвращаться на уровень выше
- ▶ Опасайтесь архитектурной жадности, надо вовремя остановиться



# На что обратить внимание

- ▶ Как представляются команды и пайплайны?
- ▶ Как создаются команды?
- ▶ Как они исполняются? Как взаимодействуют потоки в пайплайне?
- ▶ Кто и как выполняет разбор входной строки?
  - ▶ Кто, как и когда выполняет подстановки?
- ▶ Как представляются переменные окружения?
- ▶ Что с многопоточностью?