

Java

Для реализации можешь выбрать одну или несколько задач. Код пиши на java или kotlin.

Для решения третьего и четвёртого задания используй Spring.

Каждую решенную задачу сложи в отдельный репозиторий любимой системы контроля версий.

Для каждой задачи заготовь документацию — README файл, в котором опиши условия задачи, инструкции по сборке, запуску и использованию приложений.

Задание 1

I'm too young to die

Напишите приложение, которое на вход, через параметры командной строки, получит текст и выдаст статистику:

1. Количество слов в тексте
2. TOP10 самых часто упоминаемых слов, упорядоченных по количеству упоминаний в обратном порядке, по алфавиту.

Запускаться приложение должно так:

```
yourapp the quick brown fox jumps over the lazy dog
```

Результат:

В тексте 9 слов

TOP10:

2 – the

1 – brown

1 – dog

1 – fox

1 – jumps

1 – lazy

1 – over

1 – quick

** Если приложению не передано ни одного параметра, то считай список слов для сортировки из стандартного потока ввода (stdin), чтобы приложение дополнительно могло запускаться вот так:*

```
echo "the quick brown fox jumps over the lazy dog" | yourapp
```

Задание 2

Hey, not too rough

Необходимо реализовать интерфейс и покрыть код модульными тестами.

```
interface IEmployee {  
  
    // Добавить сотрудника в список.  
    // Если сотрудник с таким именем уже есть то вернуть false и не добавлять в коллекцию  
    // иначе добавить и вернуть true
```

```

fun addNewEmp(emp: Employee): Boolean

// Вернуть список 10 сотрудников, с указанной фамилией.
// Если есть несколько сотрудников с одинаковыми именем и фамилией, то
// в списке к их ФИО добавить должность
fun listEmpsBySurname(surname: String): List<Employee>

// Вернуть список 10 сотрудников с должностью, отсортированный по ФИО
fun listEmpsByPosition(position: String): List<Employee>

// Удалить по ид
fun deleteEmp(id: String)
}

```

Задание 3

Hurt me plenty

Нужно спроектировать и реализовать бэкенд (REST API) для сервиса аналогичного pastebin.com — сервис позволяет заливать куски текста/кода ("пасту") и получать на них короткую ссылку, которую можно отправить другим людям.

При загрузке "пасты" пользователь указывает:

1. Срок в течение которого "паста" будет доступна по ссылке (expiration time)
10 мин., 1 час, 3 часа, 1 день, 1 неделя, 1 месяц, без ограничения
после окончания срока получить доступ к "пасте" нельзя, в том числе и автору
2. Ограничение доступа:
public — доступна всем
unlisted — доступна только по ссылке

Для загруженной пасты выдается короткая ссылка вида

<http://my-awesome-pastebin.tld/{какой-то-рандомный-хэш}>, например,
<http://my-awesome-pastebin.tld/ab12cd34>

Пользователи могут получить информацию о последних 10 загруженных публичных "пастах".

Задание 4

Ultra-Violence

Реализуй одно или несколько дополнительных требований для третьего задания:

1. Авторизация/регистрация
 1. По логину/паролю
 2. Через одну из соц. сетей
2. Управление своими "пастами": пользователь может получить информацию о всех своих "пастах" и изменять уровень доступа к ним, включая новый уровень "private" — "паста" с таким уровнем доступна только автору
3. Поиск по названию и/или тексту "паст", с учётом ограничений доступа
4. Покрой код модульными и интеграционными тестами > 50%
5. Упакуй решение в Docker-контейнер

