

# ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3. “ЗАДАЧИ НА МОДЕЛИ” (4ЧАСА - 2 ЗАНЯТИЯ)

## 1.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Collection/Stream API/COMPARE

<https://habr.com/en/post/224593/>

<https://habr.com/en/post/213805/>

<https://dzone.com/articles/why-we-need-lambda-expressions>

<https://dzone.com/articles/why-we-need-lambda-expressions-0>

[https://vk.com/java\\_jvm/streams?act=s&amp%3Bid=69098977&z=video-69098977\\_456239041%2F7d6513518721a90ebc%2Fpl\\_post\\_-69098977\\_407](https://vk.com/java_jvm/streams?act=s&amp%3Bid=69098977&z=video-69098977_456239041%2F7d6513518721a90ebc%2Fpl_post_-69098977_407)

## 1.2. ФОРМУЛИРОВКА ЗАДАЧИ

При написании приложения обязательно использовать Java Code Convention. Для логгирования код должен использовать log4j или log4j2.

Познакомится с базовыми структурами данных. ArrayList, LinkedList, HashMap.

Далее после вводной части про StreamAPI, выполнить задание:

1. Создать 2 массива: classA[] aArray and classB[] bArray. Элементы любого из двух массивов должны содержаться в соседнем массиве.
2. см. Варианты
3. Нужно использовать peek для отладки промежуточных состояний.
4. Использовать parallelStream в подзадаче №3. Сравнить производительность с и без parallelStream.
5. Подготовить как минимум 3 примера с Optional. (Использовать orElseThrow, if(is)present, get, orElse)

Дополнительно:

- можно сделать тоже самое, но не используя Stream
- можно сделать свой коллектор

## 1.3. ВАРИАНТЫ

### 1.3.1. Вариант 1 (Библиотека):

Создать стрим Book и:

- проверить, есть ли книги, у которых больше 200 страниц;
- Найти книги с максимальным и минимальным количеством страниц
- Отфильтровать книги с единственным автором
- Отсортировать книги по количеству страниц и названию
- Получить список авторов
- Напечатать авторов, используя foreach

- Получить список авторов без дубликатов
  - a. ClassA - Author с полями:
    - short age
    - List<Book> books
  - b. ClassB - Book с полями:
    - String title
    - List<Author> authors
    - Int numberOfPages

### 1.3.2. Вариант 2 (Интернет магазин):

Создать стрим Item и:

- проверить, есть ли товары, дороже 500;
- Найти товары с максимальным и минимальным количеством единиц на складе
- Отфильтровать товары с единственным магазином
- Отсортировать товары по цене и количеству
- Получить список магазинов
- Напечатать информацию о магазинах, используя foreach
- Получить список магазинов без дубликатов
  - a. ClassA - Item с полями:
    - String name
    - Int price
    - Int count
    - List<Store> stores
  - b. ClassB - Store с полями:
    - String name
    - List<Item> items
    - List<String> feedbackList

### 1.3.3. Вариант 3 (Бронирование):

Создать стрим Aircraft и:

- проверить, есть ли самолеты вместимостью больше 200;
- Найти самолеты с максимальной и минимальной вместимостью
- Отфильтровать самолеты, которые есть у единственной авиакомпании
- Отсортировать товары по цене и вместимости
- Получить список авиакомпаний
- Напечатать информацию о авиакомпаниях, используя foreach
- Получить список авиакомпаний без дубликатов
  - a. ClassA - Aircraft с полями:
    - String name
    - Int price
    - Int capacity
    - List< Owner > stores
  - a. ClassB - Owner с полями:
    - String name
    - List<Aircraft> aircraftList
    - String location

### 1.3.4. Вариант 4 (Социальная сеть):

Создать стрим User и:

- проверить, есть ли пользователи с более чем 300 постами;
  - Найти самолеты с максимальным и минимальным количеством постов
  - Отфильтровать пользователей с единственным постом
  - Отсортировать пользователей по возрасту и количеству постов
  - Получить список постов
  - Напечатать информацию о постах, используя foreach
  - Получить список постов без дубликатов
- a. ClassA - User с полями:
    - String name
    - Int age
    - List< Post> posts
  - b. ClassB - Post с полями:
    - String content
    - List<User> likedByList

### 1.3.5. Вариант 5 (Столовая):

Создать стрим Dish и:

- проверить, есть ли блюда дороже 10
  - Найти блюда с максимальной и минимальной ценой
  - Отфильтровать блюда, с единственным покупателем
  - Отсортировать товары по популярности
  - Получить список клиентов
  - Напечатать информацию о клиентах, используя foreach
  - Получить список клиентов без дубликатов
- a. ClassA - Dish с полями:
    - String name
    - Int price
    - List<Enum> componentList
    - List< Client> clientList
  - b. ClassB - Client с полями:
    - String name
    - List<Dish> lunch
    - String location