

1. Есть два IntStream.

Первый IntStream содержит четные числа, а второй IntStream содержит нечетные числа.

Создайте третий поток, содержащий числа из обоих потоков, которые делятся на 3 и 5.

вернуть IntStream из метода шаблона, а не список целых чисел.

Два первых подходящих числа в отсортированном списке необходимо пропустить.

30, 75, 60, 90 -> 75, 90

2. **Censor**

Напишите программу, которая будет обнаруживать плохие слова в тексте в соответствии со списком плохих слов.

Все слова в тексте разделяются пробелам

Вывести список плохих слов в тексте.

Результат должен быть отсортирован, и слова не должны повторяться.

3. **Factorial**

Найти факториал числа

*Используйте шаблон:*

```
public static long factorial(long n) {  
}
```

4. Напишите метод вычисления суммы нечетных чисел в заданном интервале (включительно) с помощью Stream API.

*Используйте шаблон:*

```
public long sumOddNumbersBetween(int a, int b) {  
    return ...  
}
```

5. **Transaction**

У вас есть два класса:

Transaction:

**uuid:** String, **state:** State (CANCELED, FINISHED, PROCESSING), **sum:** Long, **created:** Date

Account:

**number:** String, **balance:** Long, **transactions:** List<Transaction>

Напишите метод, использующий stream API, который вычислит общую сумму в размере погашенных сделок для всех непустых счетов ( **balance** > 0 ).

*Используйте шаблон:*

```
public class AccountService {  
    public long getSumAllCancelledTransactions(List<Account> accounts) {  
        return  
    }  
}
```

6. **Employee**

У вас есть два класса:

Employee (name: String, salary: Long)

Department (name: String, code: String, employees: List<Employee>).

Напишите метод с использованием Stream API, который вычисляет общее количество сотрудников с зарплатой больше чем порог(threshold) каждого отдела, код которых(отделов) начинается со строки «111-» (без «»).

*Используйте шаблон:*

```
public static long calcNumberOfEmployees(List<Department> departments, long threshold) {  
    return  
}
```

