

Функциональное программирование в Java. Часть 2. Stream API

Даниил Пилипенко

Директор центра подбора
IT-специалистов SymbioWay

Skillbox

Введение в Stream API

Введение

Коллекции в Java имеют очень важное значение. Почти каждое Java-приложение создаёт и обрабатывает коллекции. Они имеют фундаментальное значение для многих задач программирования: они позволяют группировать и обрабатывать данные.

Например, вам нужно создать коллекцию строк. Затем вы захотите обработать коллекцию, чтобы отобрать только те строки, которые не являются числами. Либо вам нужно подсчитать сумму символов в таких строках.

Для реализации всех этих примеров необходимо выполнить ряд действий. В первую очередь необходимо организовать цикл, который перебирает коллекцию. Затем написать условие, которое проверяет элементы коллекции, либо какой-то код, который производит с ними те или иные действия.

Императивный ПОДХОД

Императивным называется такой подход, когда программист на каждом шагу контролирует работу программы и явно прописывает, каким образом совершать те или иные действия.

```
private int getResult_imperative(List<String> strings) {  
    int result = 0;  
    for (String string : strings){  
        if(isDigit(string.charAt(0))) {  
            continue;  
        }  
        if (string.contains("_")) {  
            continue;  
        }  
        result += string.length();  
    }  
    return result;  
}
```

Stream API

В JDK 8 был представлен новый подход для работы с коллекциями — Stream API.

Это библиотека, которая позволяет работать с коллекциями и подобными им источниками данных в декларативном стиле.

Декларативный подход

Если вызовы методов объединяются в цепочку последовательно, то такой подход называется декларативным от слова declare — объявлять.

```
private int getResult_Declarative(List<String> strings) {  
    return strings  
        .stream()  
        .filter(s -> !isDigit(s.charAt(0)))  
        .filter(s -> !s.contains("_"))  
        .mapToInt(String::length)  
        .sum();  
}
```

Вывод

В следующих видео вы погрузитесь глубже в Stream API, разберёте его методы и различные сценарии использования.