Функциональное программирование
в Java. Часть 2. Stream API

Skillbox

Parallel, reduce и комбинирование операторов

Параллельное выполнение

Зачастую вы хотите, чтобы ваша программа выполнялась быстрее.

Однако в стриме элементы обрабатываются по очереди друг за другом.

Ускорить эту обработку вы сможете, если будете выполнять операции в стриме не последовательно, а одновременно.

Операция reduce

Выполняет агрегационные операции
на элементах стрима и возвращает
один результат.

Операция reduce

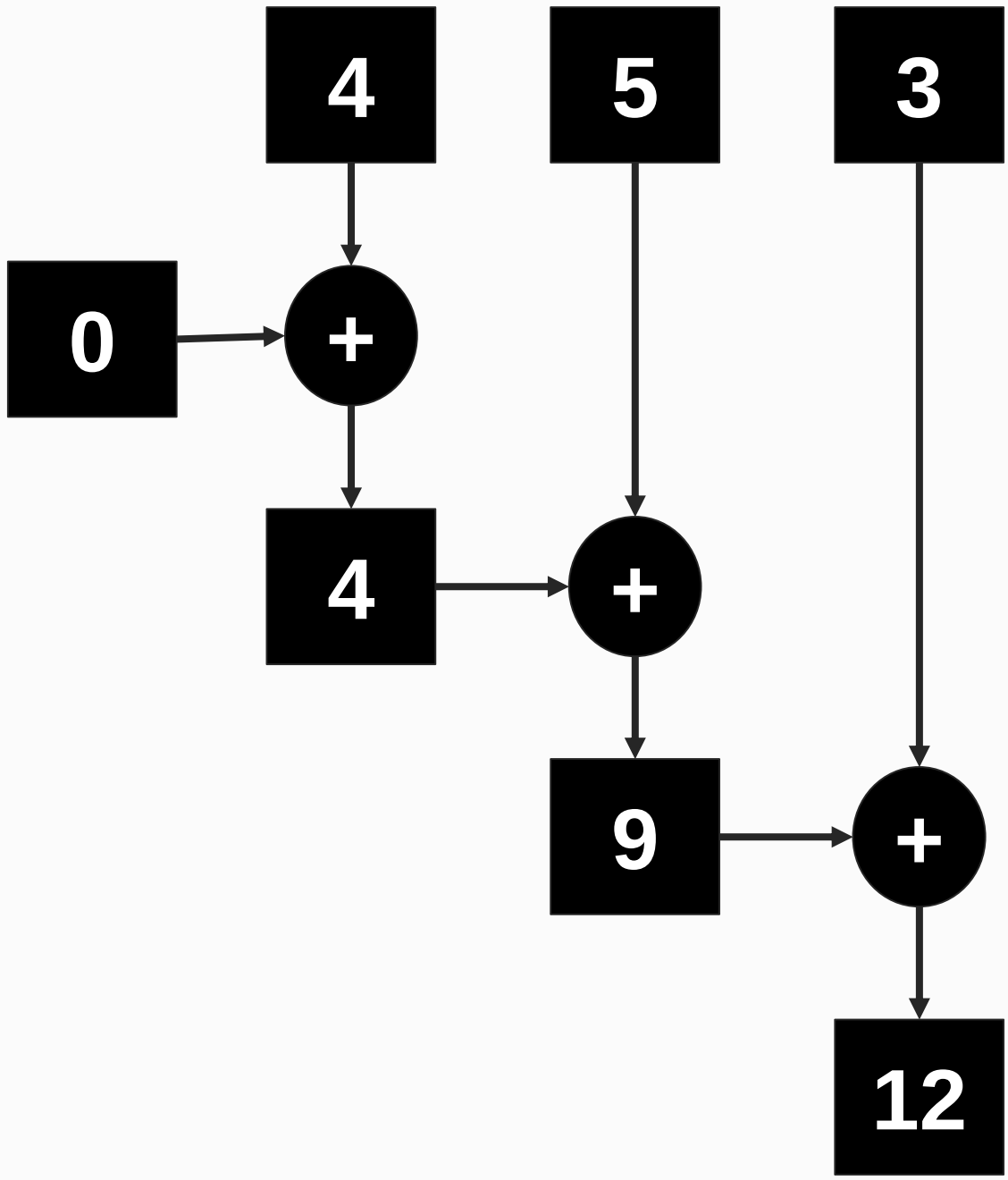

Схема: Даниил Пилипенко

Операция reduce

- **T** reduce(**T** identity, BinaryOperator<**T**> accumulator)
- identity — начальное значение (для суммы это 0)
- accumulator — функциональный интерфейс с двумя параметрами, первый — результат обработки предыдущего элемента, второй — текущий элемент

Операция reduce

Варианты применения:

- получение суммы, произведения, среднего значения и применение иных математических функций
- получение результата применения логических операторов
- поиск максимального и минимального значений
- поиск элемента, лучше других подходящего под определённый критерий

Операция reduce

```
int[] numbers = {4, 5, 3, 9};

Arrays.stream(numbers).sum();

Arrays.stream(numbers).average().orElse(0.0);

Arrays.stream(numbers).max()
    .orElse(Integer.MIN_VALUE);

Arrays.stream(numbers).min()
    .orElse(Integer.MAX_VALUE);

Arrays.stream(numbers).summaryStatistics();
```

Изображение: Даниил Пилипенко

Выводы

В этом модуле вы познакомились с механизмом Stream API, позволяющим удобно и эффективно работать с элементами коллекций и другими источниками данных.

Вы узнали о том, как создавать стримы, что такое промежуточные и терминальные операторы, как они работают и какими возможностями обладают.

Используя Stream API, вы сможете сделать ваш код более понятным, гибким и поддерживаемым, а знания Stream API помогут вам понимать и при необходимости изменять чужой код.