Adventures With Stream.reduce()

Adventurer: Dinko Srkoč

JavaCro'22 Spring

About

Dinko Srkoč

BISS d.o.o.

- Software engineer
- 20+ years experience
- FER, 1998

reduce aka fold

`Stream#reduce(...)`

[java doc]

`Stream`:

> *A sequence of elements supporting sequential and parallel aggregate operations*

`reduce()`:

> *Combines a sequence of elements into a single summary result by repeated*
> *application of a combining operation*

reduce aka fold

Sum of integers

```
final var xs = List.of(1, 2, 3, 4, 5, 6, 7);

var sum = 0;
for (var x : xs) {
    sum += x;
}

return sum;
```

Max volume

```
final var cubes = List.of(new Cube(3, 4, 2), …);

var maxCube = new Cube(0, 0, 0);
for (var cube : cubes) {
    if (cube.volume() > maxCube.volume())
        maxCube = cube;
}

return maxCube;
```

reduce aka fold

Sum of integers

```
final var xs = List.of(1, 2, 3, 4, 5, 6, 7);

var sum = 0;
for (var x : xs) {
    sum += x;
}

return sum;


xs.stream().reduce(0, Integer::sum)
```

Max volume

```
final var cubes = List.of(new Cube(3, 4, 2), …);

var maxCube = new Cube(0, 0, 0);
for (var cube : cubes) {
    if (cube.volume() > maxCube.volume())
        maxCube = cube;
}

return maxCube;


xs.stream().reduce(
    new Cube(0, 0, 0),
    (max, cube) -> cube.volume() > max.volume() ? cube : max)
```

reduce aka fold