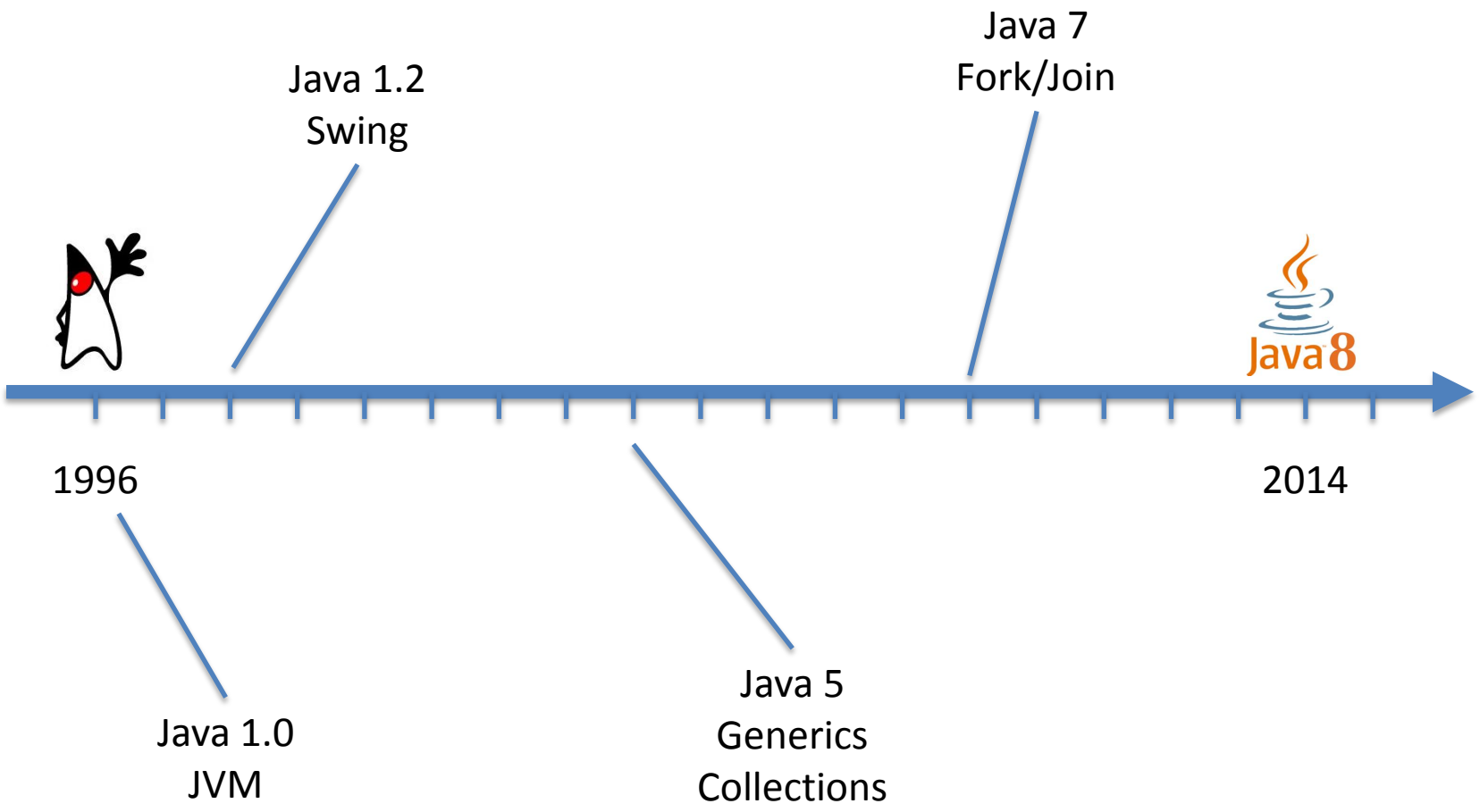
EXPLORE NEW PERSPECTIVES

Parallel and concurrent programming in Java 8

Part 0 - Introduction

Java evolution



Alignment with language trends!

Java ecosystem

Improvements in Java 8

Functional style of programming

Collection enhancements

Stream processing

Optional values

Lambda expressions

Method references

Default methods

Change the way of thinking!

Imperative programming

style of programming modeled as a sequence of commands that modify state

Functional programming

programs are expressions and transformations, modeling mathematical formulas

Change the way of thinking!

```
count = 0;
for(i = 0; i < n; i++)
    if (a[i] > 0)
        count++;
```

$$/+ \circ \alpha(>\circ[id,\bar{0}] \rightarrow \bar{1}\ ;\bar{0})$$

programming means tell —declaratively—*what* we want rather than *how* to do it.

Imperative approach

```
count = 0;
for(i = 0; i < n; i++)
    if (a[i] > 0)
        count++;
```

i | 0

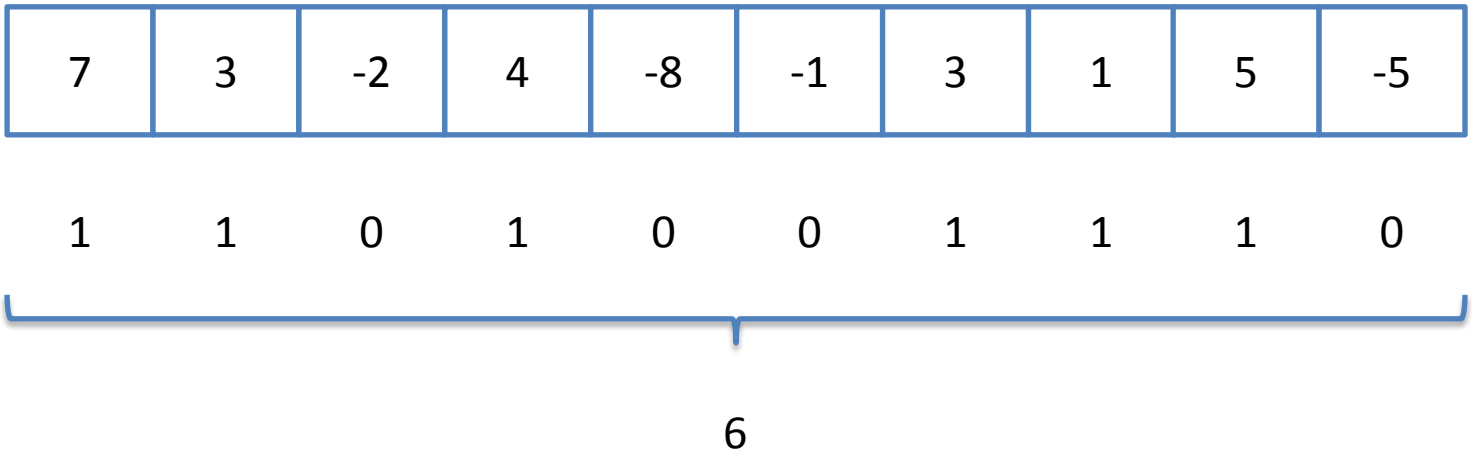
count | 0

a

7	3	-2	4	-8	-1	3	1	5	-5

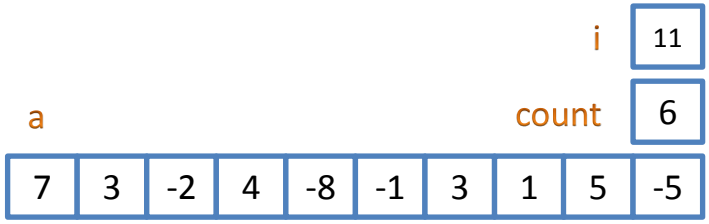
Functional approach

$$/+ \circ \alpha(> \circ [id, \bar{0}] \rightarrow \bar{1}\ ; \bar{0})$$

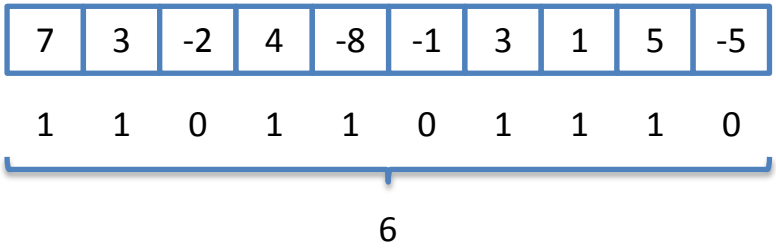


What do you think of...?

```
count = 0;
for(i = 0; i < n; i++)
   if (a[i] > 0)
      count++;
```



$$/+ \circ \alpha(> \circ [id, \bar{0}] \rightarrow \bar{1}\ ; \bar{0})$$



parallelism

mutable objects

what do you think?

different approach: what vs. how

what happen if we call twice a function?

What about Object Oriented Programming?

```
class A {
  int x;
  int getX();
  void setX(int x);
}
```

abstracting over data

$$f(g(x))$$

abstracting over behavior

Functional programming

Is it new?

1930 - Lambda Calculus (A. Church)
1958 - Lisp (J. McCarthy)
...
1977 - FP (J. Backus)
...

What about Java 8 implementation?

- no monads
- reduced lazy evaluation
- little support for immutability

...

better than nothing!

Benefits

- Simpler, cleaner, and easier-to-read code
- Simpler maintenance
- Great for collections!
- Enhanced parallelism/concurrency for multi-core CPUs


ESTECO
EXPLORE NEW PERSPECTIVES
Thank you
for your attention!
www.esteco.com