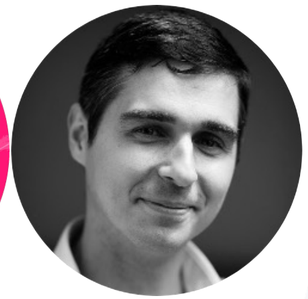
Looking Closely at Strings and Numbers



Andrejs Doronins
Software Developer in Test

Overview

- When to use StringBuilder
- Cleaner code with Text Blocks
- Syntax separator for large numbers
- Careful with floating-point numbers arithmetic

String Immutability

```
String name = "James";


String s = "Name: " + name;
String s2 = String.format("Name: %s", name);
```

New strings - new objects

Strings in Loops

```
String str = "";

for (int i = 0; i < stringArr.length ; ++i) {
  str = str + stringArr[i];
}
```

Strings in Loops

```
StringBuilder sb = new StringBuilder();

for (int i = 0; i < stringArr.length; ++i) {
    sb.append(stringArr[i]);
  }

String str = sb.toString();
```

Prefer Text Blocks

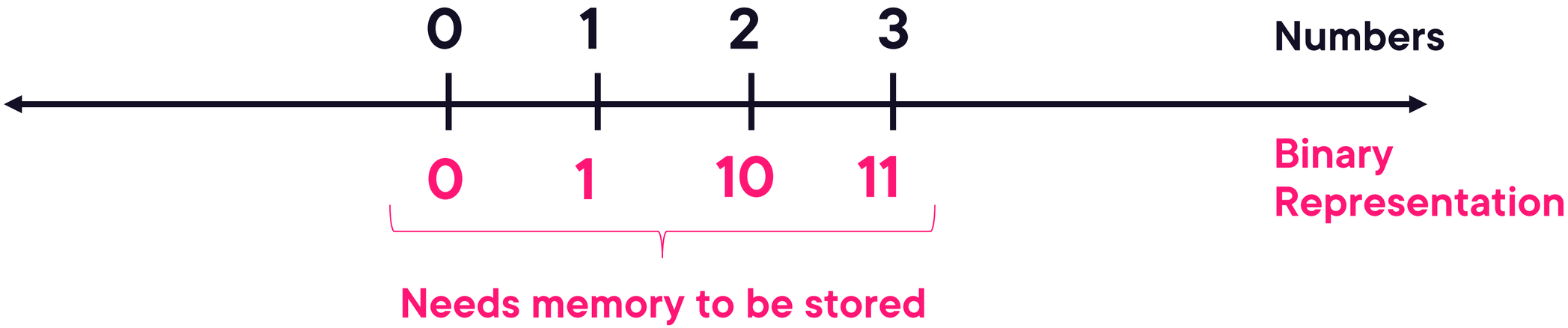
```
String json = "[\n" +
              "   {\n" +
              "     \"from\":\"New York\",\n" +
              "     \"to\":\"London\",\n" +
              "     \"date\":\"2022-07-07\"\n" +
              "   }\n" +
              "]";
```

Prefer Text Blocks

```
String json = """
              [
                {
                  "from":"New York",
                  "to":"London",
                  "date":"2022-07-07"
                }
              ]
              """;
```

Use Separators

```
int num  = 100;
int num2 = 1000000;   // million?
int num3 = 1_000_000; // million!
```


0
1
2
3
Numbers
0
1
10
11
Binary
Representation
Needs memory to be stored

0.1
0.01
0.001
∞

- Need infinite memory to store
- Approximate representation

```
class Money {

    BigDecimal amount;
    Currency currency;
}
```

JSR 354 - Money and Currency

Be aware of issues with floating-point arithmetic

Consider the loss of precision

Prefer BigDecimal

For Money - use a dedicated API (amount + currency)

Summary

- StringBuilder in loops
- Text Blocks for JSON or other multi-line strings
- Separators for numbers
- BigDecimal for precision

Up Next:

Better Iterating and Branching