Naming Matters

Andrejs Doronins
TEST AUTOMATION ENGINEER

The importance of naming is often underestimated

```java
class CommonManager { // what does this do?

    Map<String, String> data; // what kind of data?

    public Map<String, String> getData() {
        //....
    }
}
```

Module Overview

Class Names

Variable Names

Method Names

Common Guidelines

Class Names Guidelines

Noun

- Concrete: Dog, House, Calculator
- Abstract: SalaryAlgo, EmailSender

Specific

Where Are the House Keys?

Stuff

Things

Common

Other

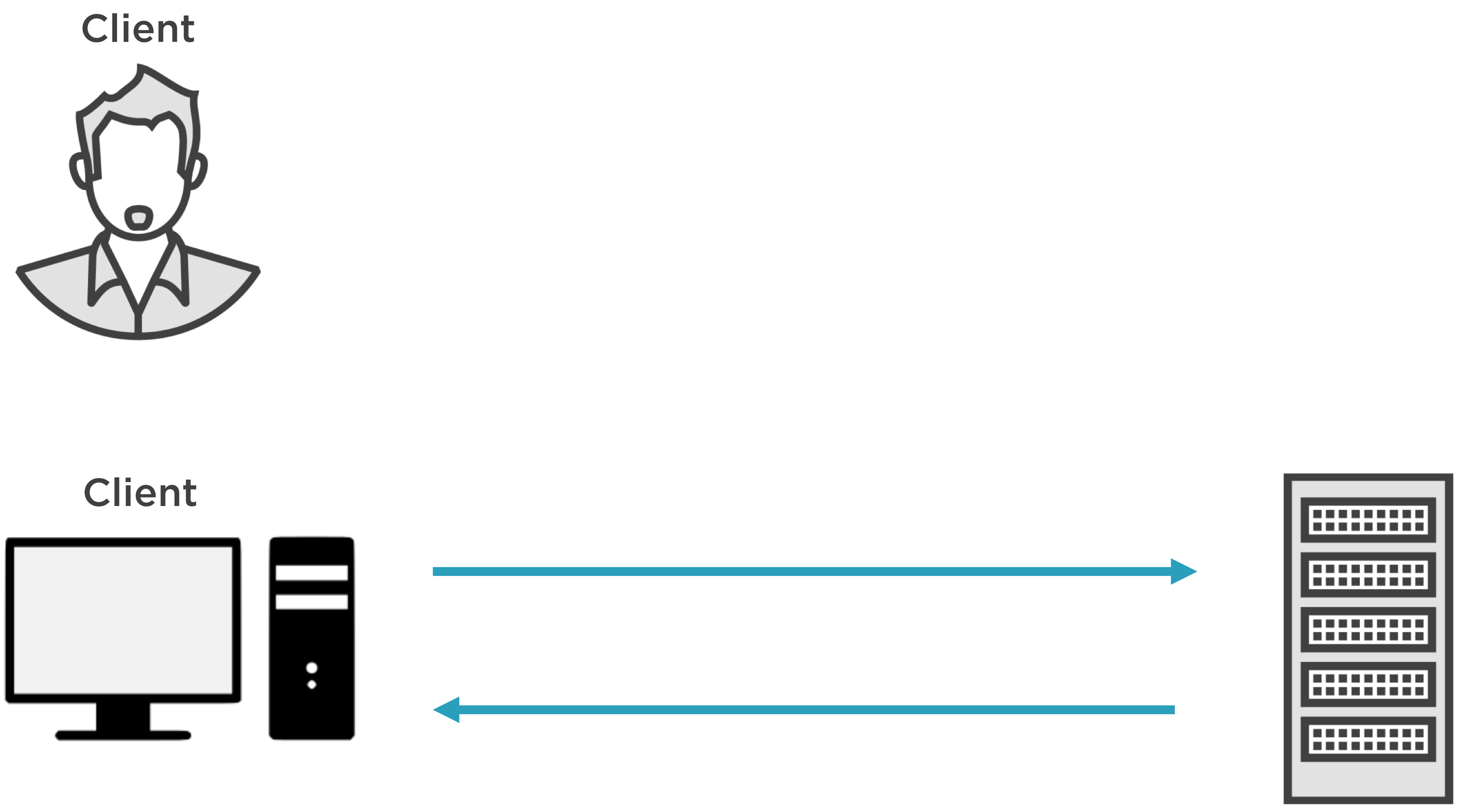
What’s in This Class?

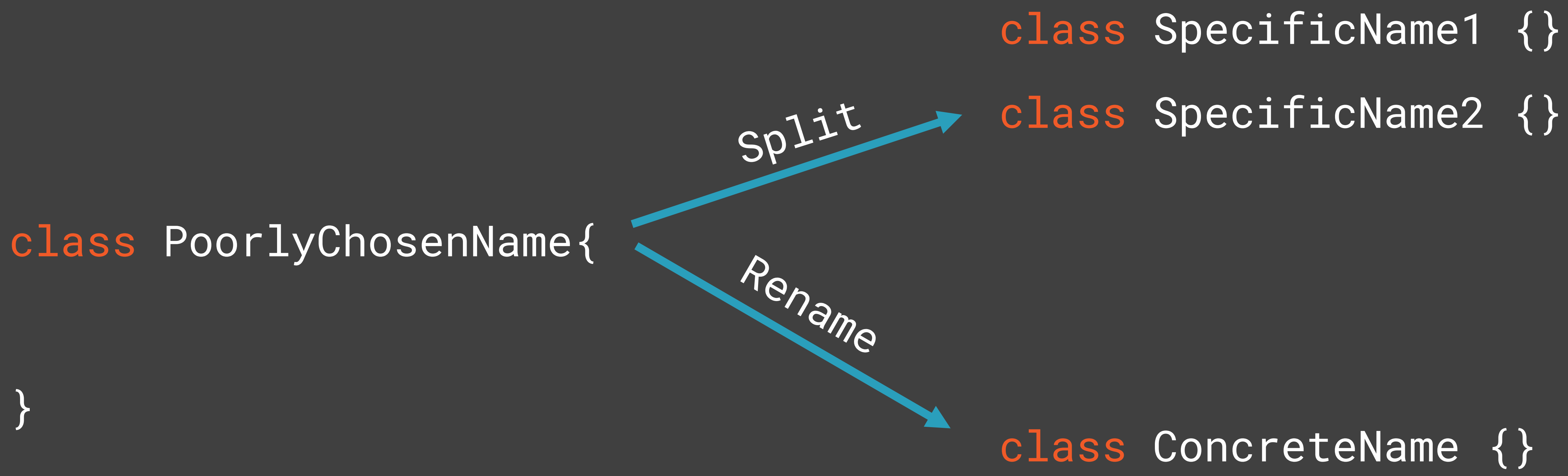
PoorlyNamedClass.java

All kind of ~~things~~ code

SRP

Single Responsibility Principle


Client
Client



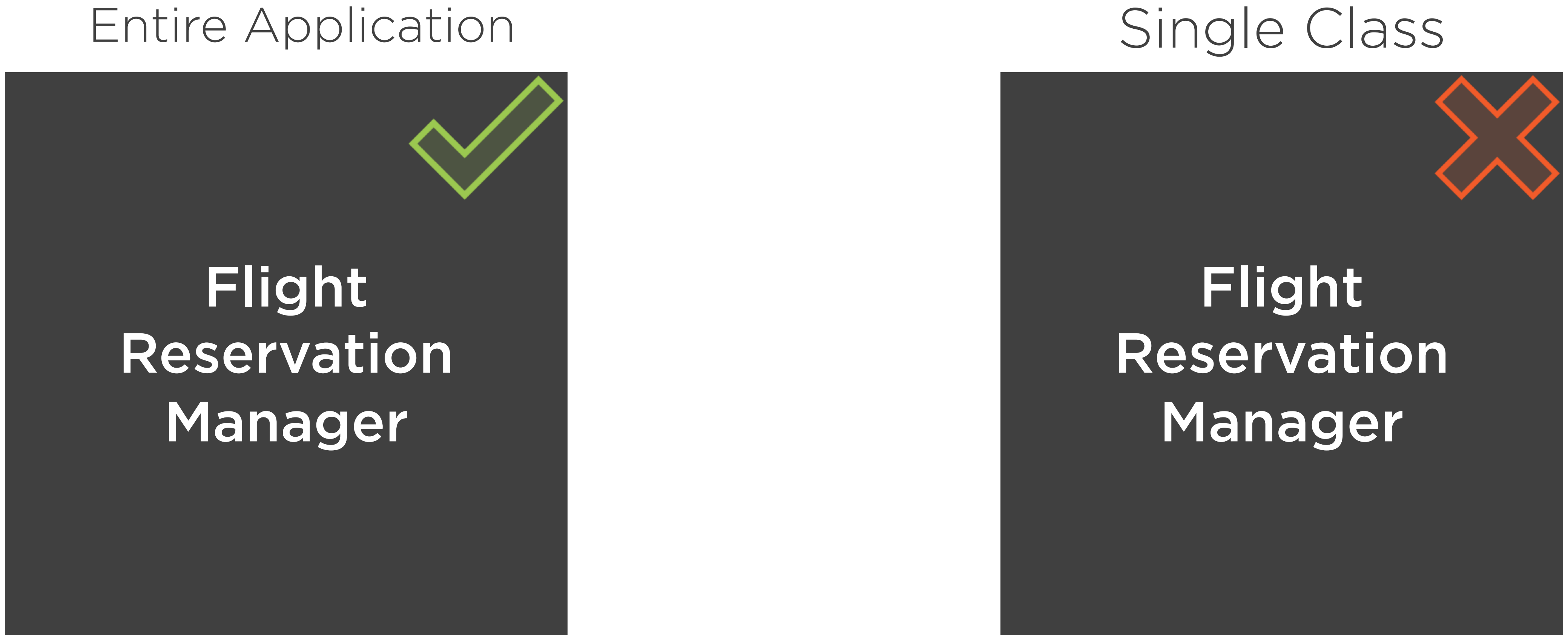
```
class PoorlyChosenName{


}
```

```
class SpecificName1 {}
class SpecificName2 {}
```

```
class ConcreteName {}
```

Avoid

- *Coordinator
- *Manager
  - e.g. StoreManager, FlightManager


Entire Application
Flight Reservation Manager
Single Class
Flight Reservation Manager

Some Alternatives

Builder

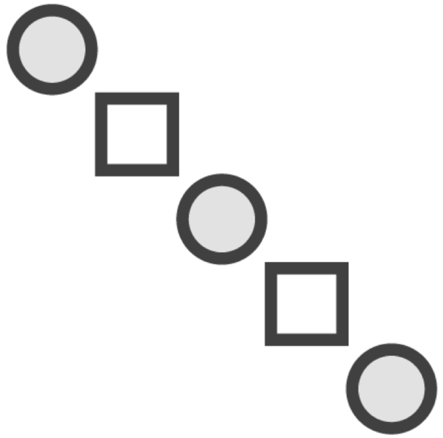
Writer

Reader

Handler

Container

Patterns



Builder

Singleton

Factory

(and other)

```
// OK, provides car objects
class CarFactory {
}

// OK, builds a client with method chaining
class HttpClientBuilder {
}
```

Don’t Get Carried Away

```
SingletonCarFactoryProxyHandler
```

Real Class from Spring Framework

```
J2eeBasedPreAuthenticatedWebAuthenticationDetailsSource
```

Variable Name Guidelines

Never a single letter

Always specific

Ideally 1-2 words

booleans prefixed with "is", for example isActive or isValid

use camelCase

use ALL_CAPS with underscores for constants

```
Map<String, String> d = getThings();                    ✗
Map<String, String> data = getThings();                 ✗
Map<String, String> customerDetails = getThings();      ✓
```

Method Name Guidelines

Should reveal intent

Functionality fully understandable from the name

If you have to look inside the method to understand what it does – the name needs improvement

Verb (Do What?)		Noun (To What?)		Result
load	+	Page	=	loadPage()
set	+	Price	=	setPrice()
convert	+	Currency	=	convertCurrency()

Be Specific!

Verb (Do What?)		Noun (To What?)		Result
load	+	~~Data~~ customerDetails	=	loadCustomerDetails()
set	+	~~Value~~ Price	=	setPrice()

```
Map<String, String> customerDetails = getThings();          ✗

Map<String, String> customerData = getCustomerData();       ✓
```

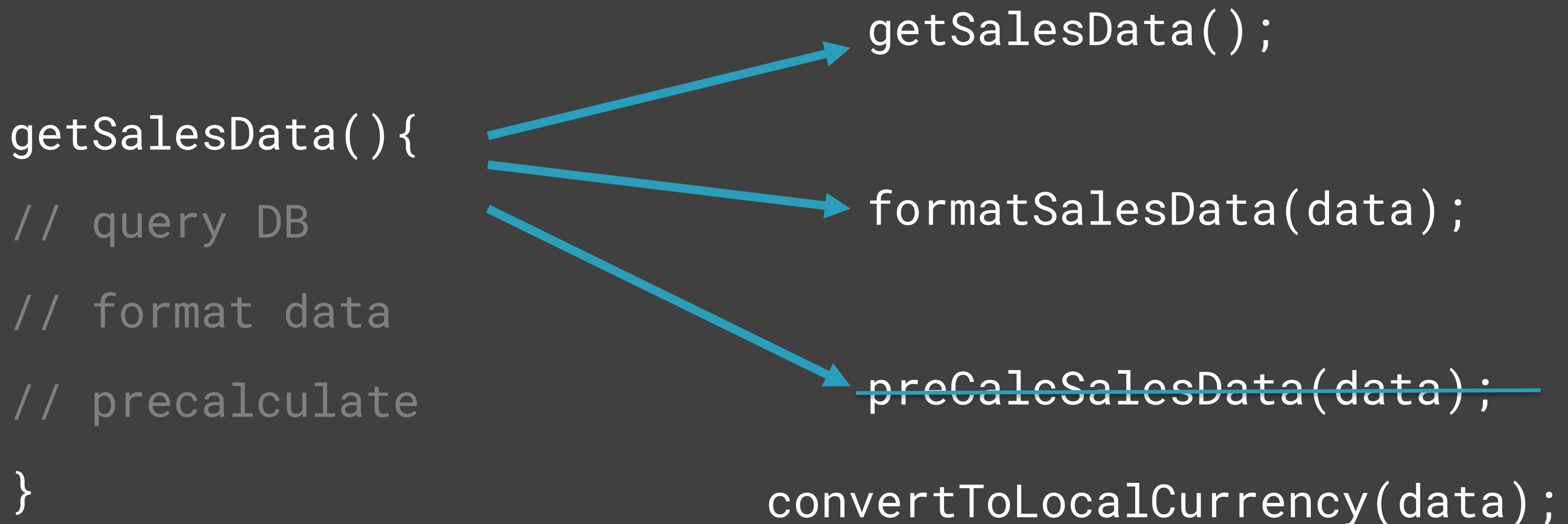
Java String Class

```
"aa".concat("b").endsWith("b");
```

Method Name Anti-patterns

Method does more than the name says

Name contains “and”, “or”, “if”



```
getSalesData(){
// query DB
// format data
// precalculate
}
```

`getSalesData();`

`formatSalesData(data);`

~~preCalcSalesData(data);~~

`convertToLocalCurrency(data);`

```
getAndFormatAndPreCalculateSalesData(){ ✗
// query DB
// format data
// precalculate
}
```

Breaking Method Name Rules

Static Factory Methods

Builder and Fluent Interface patterns

Java 8 Streams

```
someList.stream()
        .map(func1.andThen(func2))
        .findAny()
        .orElseThrow(...);
```

Pls, dnt use abbrvtns

Universal Abbreviations?

kg

km

lbs -------

What are labs?

Careful with typos and spelling

Summary

✓ Classes – Single Responsibility

✓ Variables – descriptive and concise

✓ Methods – reveal intent and no multi-tasking