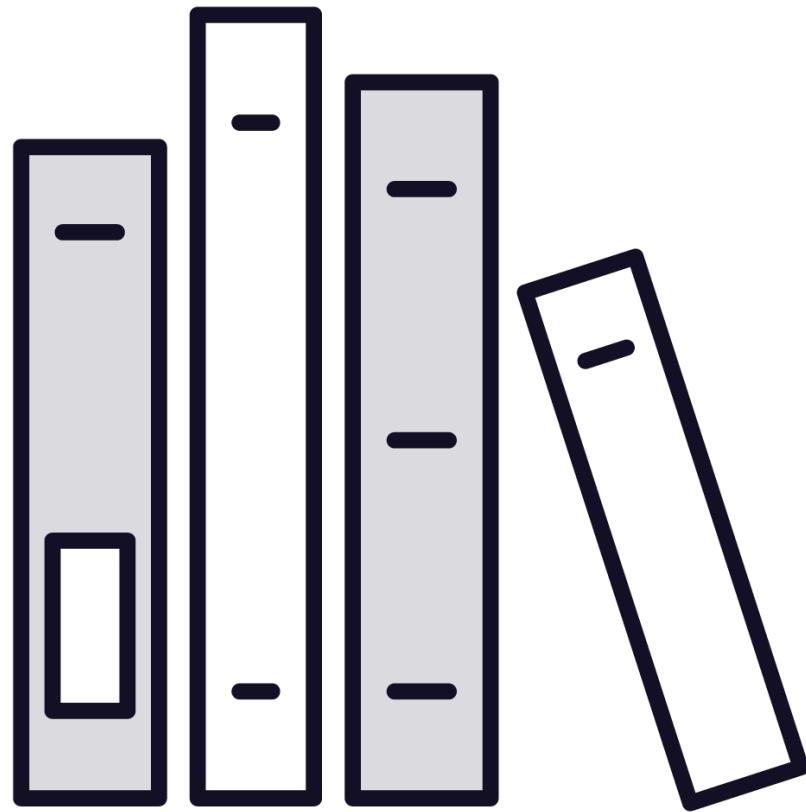


What's in a Name?



Andrejs Doronins
Software Developer in Test



Book

Chapter 1

Chapter 2

Classes

Methods

Variables



```
class CommonManager { // what does this do?  
    Map<String, String> data; // what kind of data?  
  
    public Map<String, String> getData() {  
        // ....  
    }  
}
```

Naming - easy to understand, hard to get right

**If you misname things, you
will misuse them**



Overview

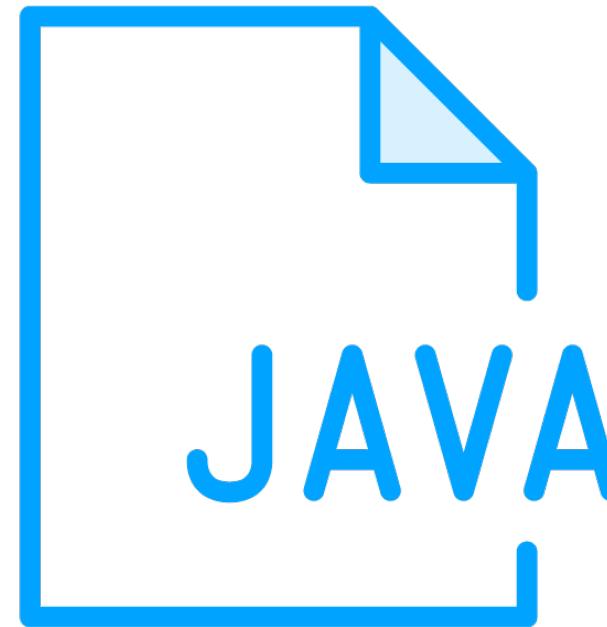


Naming matters!

- classes
- variables
- methods
- exceptions to the rules



Class Names Guidelines



Noun

- Concrete: Dog, House, Airport
- Abstract: SalaryAlgo, Booking, Flight

Specific

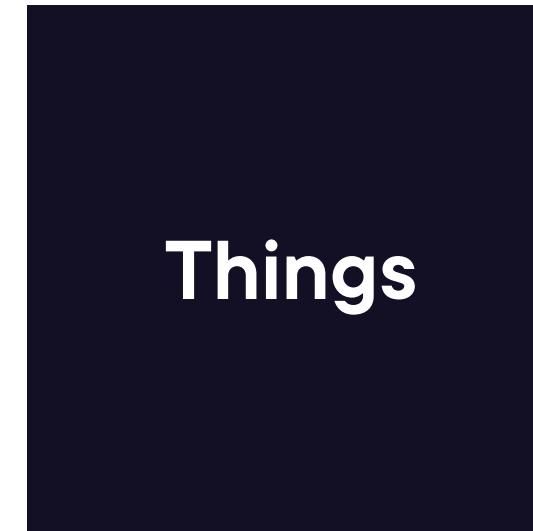


Where Are the House Keys?

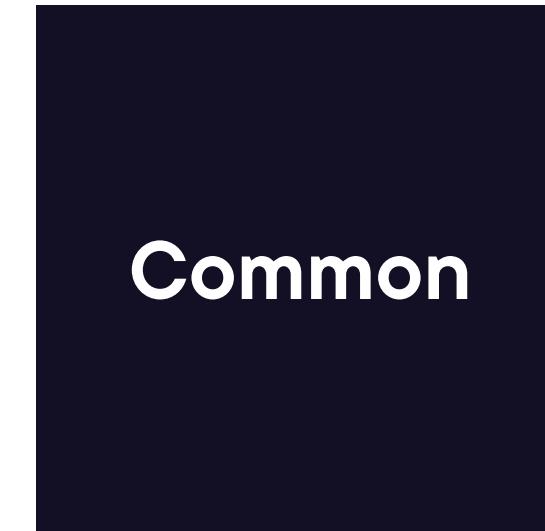
Container.java



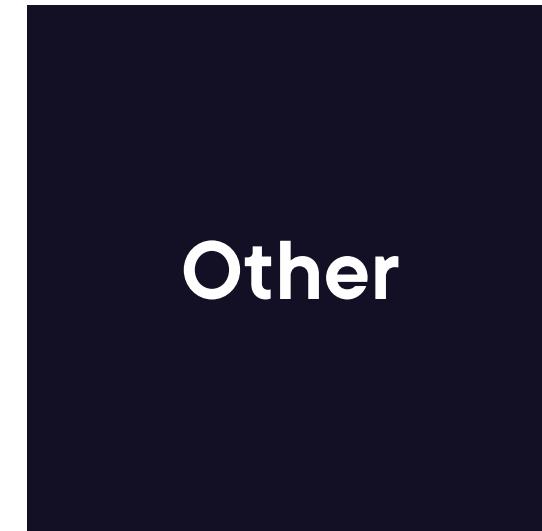
Data.java



CommonHandler.java



OtherData.java



Stuff

Things

Common

Other



SRP

Single Responsibility Principle



VaguelyNamedClass.java



All kind of
things code



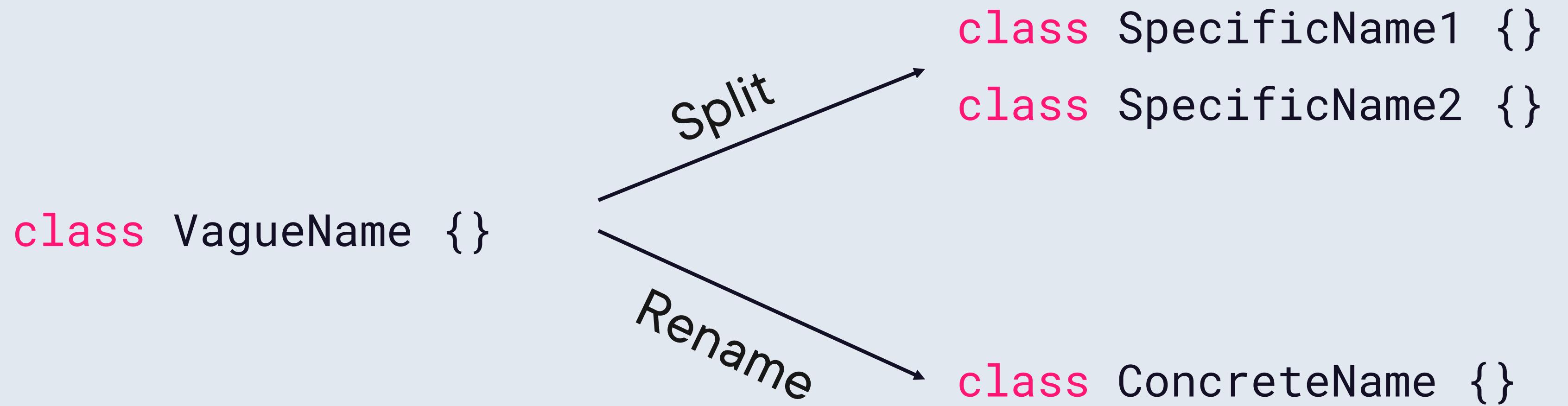
VaguelyNamedClass.java

All kind of
things code
... and more
... and then some

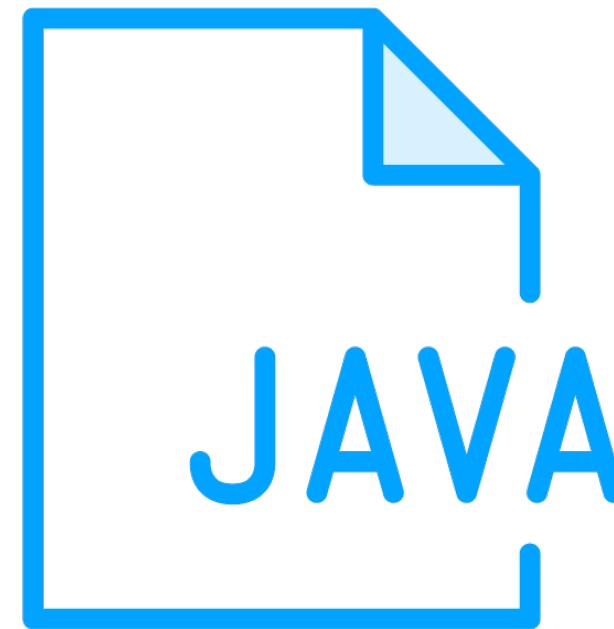


```
class AirlineManager {  
    List<Flight> searchFlights();  
  
    void viewBookings();  
  
    void displayFleetInfo();  
  
    StaffData getStaffData();  
}
```

Breaks the SRP



Variable Names Guidelines



Never a single letter

Always specific

Ideally 1-2 words

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

use camelCase

use ALL_CAPS with underscores for constants



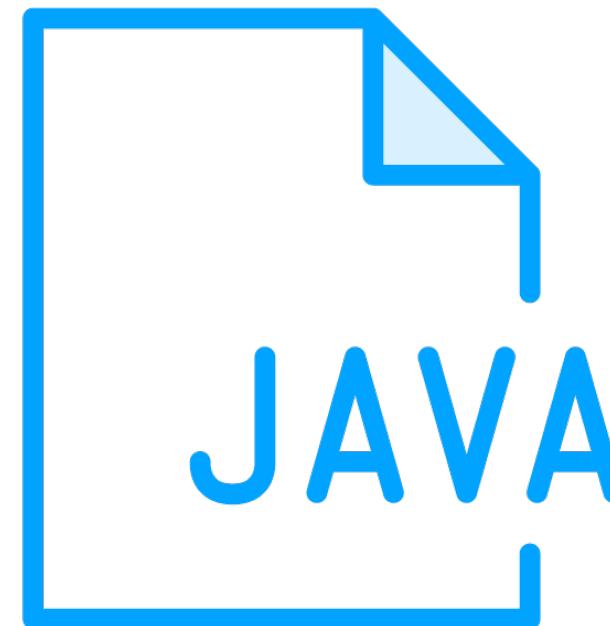
```
Map<String, String> d = getThings();
```

```
Map<String, String> data = getThings();
```

```
Map<String, String> passengerDetails = getThings();
```



Method Names Guidelines



Specific

Ideally 1-2 words

"is" or "has" if it returns a boolean

Should reveal intent

Purpose understandable from the name



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



Be Specific!

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

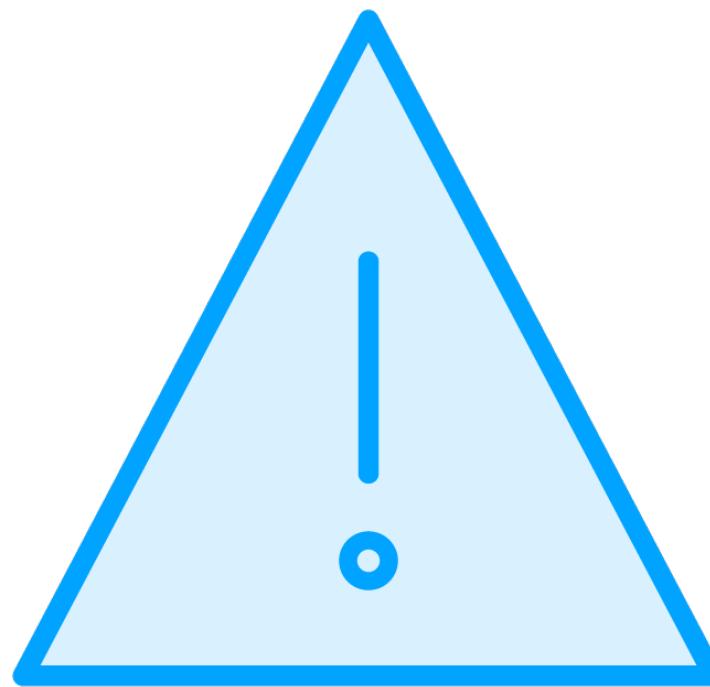


```
Map<String, String> passengerData = getThings();
```

```
Map<String, String> passengerData = getPassengerData();
```



Method Antipattern

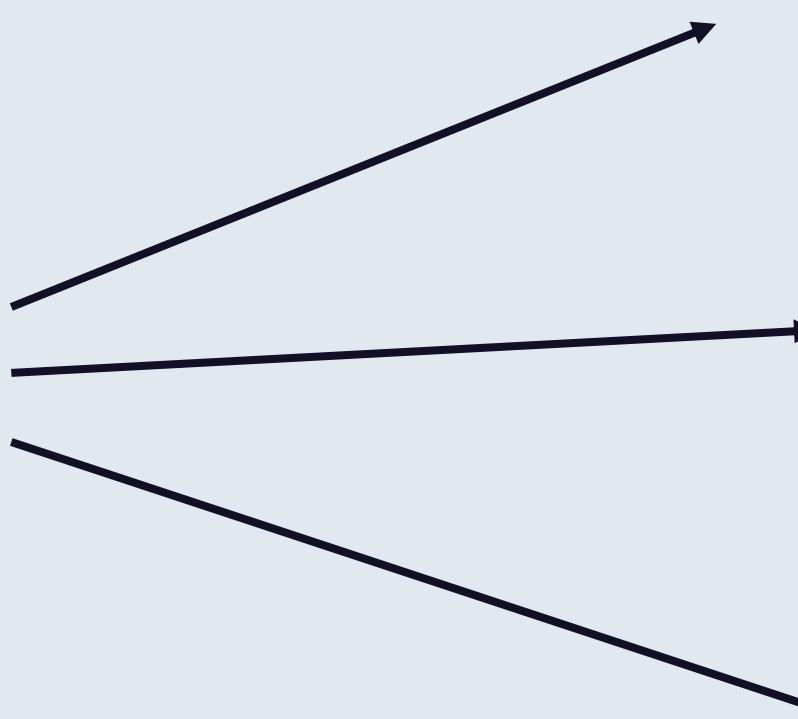


Your method implementation:

- “this method does A and B”
- does multiple things
- should be split



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



getSalesData();

formatSalesData(data);

~~preCalcSalesData(data);~~

convertToLocalCurrency(data);

getAndFormatAndPreCalculateSalesData()

Naming

**Smaller
methods**

**Reusable
Code**



Naming exceptions exist



Java String

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



Java Streams

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



```
record Flight(String from, String to) {  
}
```

```
var flight = new Flight("London", "New York");  
flight.from(); ←  
flight.to(); ←
```



Breaking Method Name Rules



Static Factory Methods
Builder pattern
– Course: Java Design Patterns



Pls, dnt use abbrvtns



Universal Abbreviations?



kg

km

lbs

What are labs?



Industry Specific Abbreviations



Investment banking (trading):

- price (px)
- quantity (qty)
- currency (ccy)

OK if known to virtually everyone



Carful with typos and speling

(it affects searching)



Summary



Classes:

- Noun, specific
- Split large classes
- Helps adhere to SRP

Variables:

- 1-2 words, precise

Methods:

- Intent-revealing verbs
- Split methods that do many things



Up Next:

Creating Objects the Right Way

