

Better Constructors



Andrejs Doronins

TEST AUTOMATION ENGINEER



Module Overview



Static Factory Methods

Constructor Chaining

Builder Pattern



```
new GregorianCalendar( new TimeZone() , new Locale())
```



“Consider static factory methods instead of constructors”

Joshua Bloch



Static Factory Methods in Java

```
Calendar.getInstance();
```

```
String.valueOf(true);
```

```
LocalDate.of(2019, 01, 01);
```

```
Optional.empty();
```

```
Collections.unmodifiableCollection(...);
```



Static factory methods
encapsulate the
construction logic



DRY
Don't Repeat Yourself





Constructor chaining helps
adhere to the DRY principle



Telescoping Constructor Anti-pattern

```
Pizza(int size) { ... }
```

```
Pizza(int size, boolean cheese) { ... }
```

```
Pizza(int size, boolean cheese, boolean ham) { ... }
```

```
Pizza(int size, boolean cheese, boolean ham, boolean mushroom) { ... }
```



Using the Builder Pattern

```
Hero mage = HeroBuilder(Mage)
    .life(100)
    .magic(200)
    .build();
```

```
Car car = CarBuilder(Ford)
    .color("Black")
    .year(2018)
    .build();
```



Summary



Static Factory Methods



Constructor chaining



Builder and Fluent Interface

