

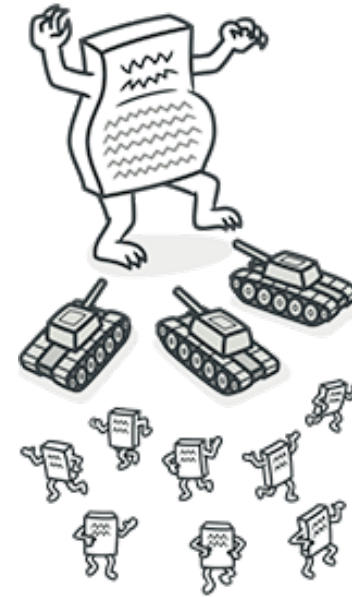
MARCO GRÜTER 22.03.2022

# REFACTORING RECAP

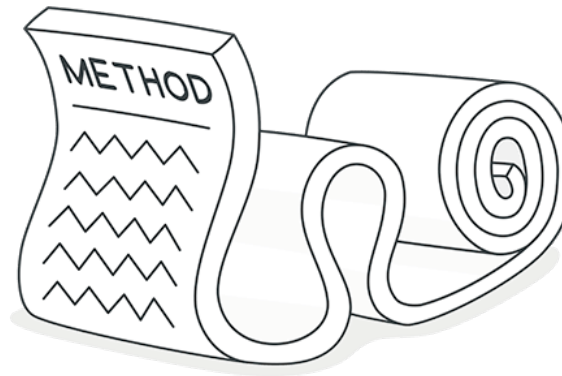
Code smells - Bloaters

# BLOATERS

- Long methods
- Large class
- Primitive Obsession
- Long Parameter List
- Data Clumps



# LONG METHOD



Too many lines of code



> 10 lines of code

# LONG METHOD - TREATMENT 1



**Extract Method**

# LONG METHOD - TREATMENT 2



Extract Method



**Replace Temp with Query**

# LONG METHOD - TREATMENT 3



Extract Method

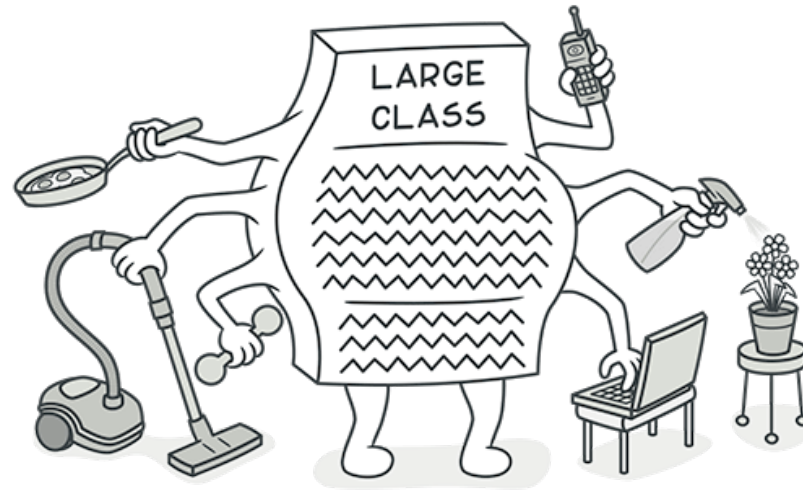


Replace Temp with Query



**Decompose Conditional**

# LARGE CLASS

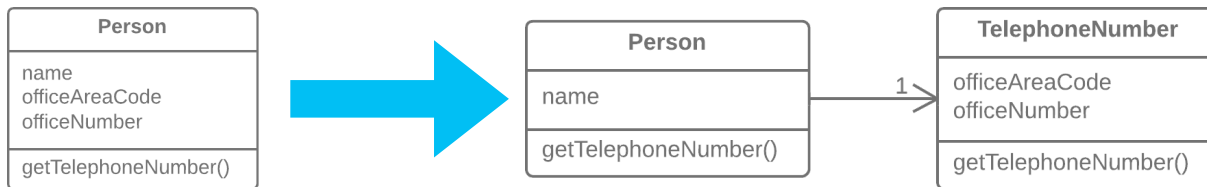


Too many

- fields
- methods
- lines of code

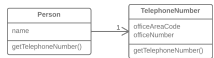
# LARGE CLASS - TREATMENT 1

## Extract Class



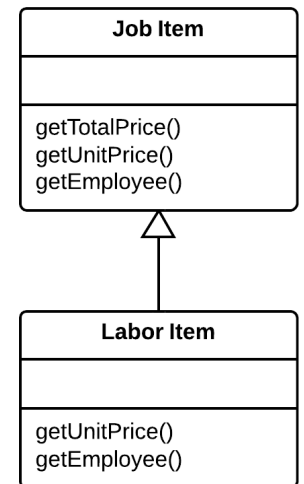
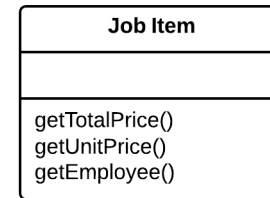


# LARGE CLASS - TREATMENT 2

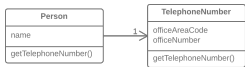


Extract Class

Extract Subclass



# LARGE CLASS - TREATMENT 3

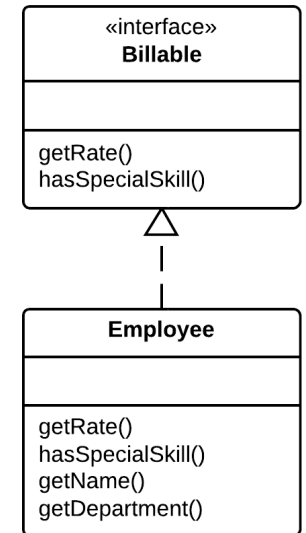
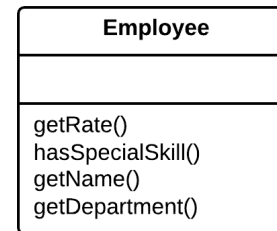


Extract Class

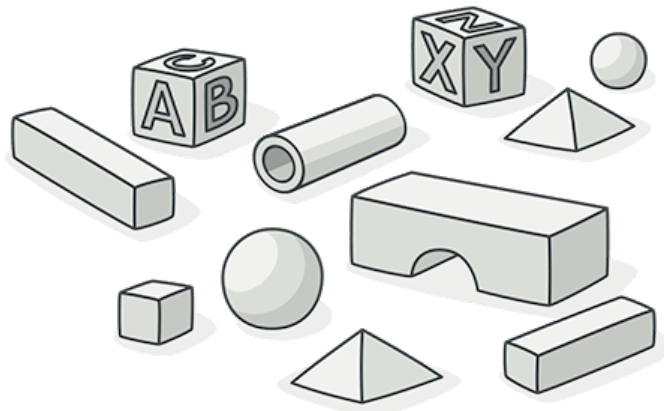


Extract Subclass

Extract Interface



# PRIMITIVE OBSESSION



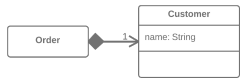
Primitives vs. small objects

# PRIMITIVE OBSESSION - TREATMENT 1

## Replace Data Value with Object

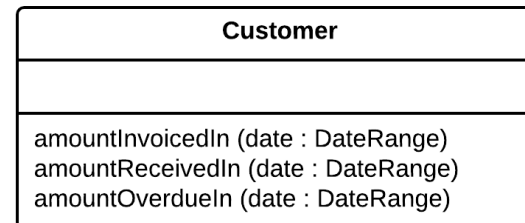
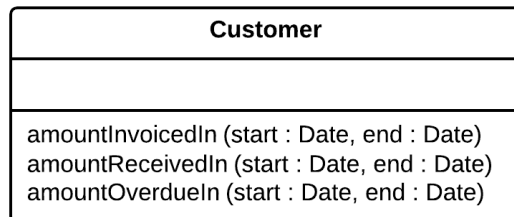


# PRIMITIVE OBSESSION - TREATMENT 2



Replace Data Value with Object

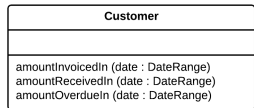
## Introduce Parameter Object



# PRIMITIVE OBSESSION - TREATMENT 3



Replace Data Value with Object



Introduce Parameter Object

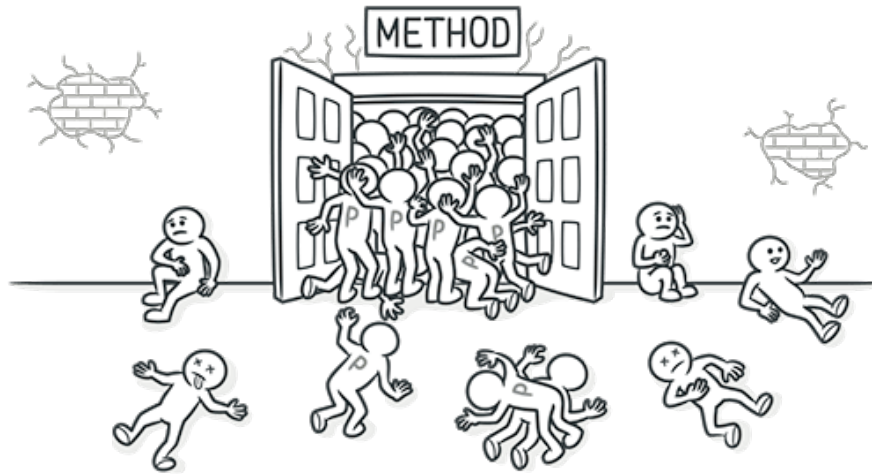
## Preserve Whole Object

```
int low = daysTempRange.getLow();
int high = daysTempRange.getHigh();
boolean withinPlan = plan.withinRange(low, high);
```



```
boolean withinPlan =
    plan.withinRange(daysTempRange);
```

# LONG PARAMETER LIST



> 3 parameters for a method

# LONG PARAMETER LIST - TREATMENT 1

## Replace Parameter with Method Call

```
int basePrice = quantity * itemPrice;  
double seasonDiscount = this.getSeasonalDiscount();  
double fees = this.getFees();  
double finalPrice =  
    discountedPrice(basePrice, seasonDiscount, fees);
```



```
int basePrice = quantity * itemPrice;  
double finalPrice =  
    discountedPrice(basePrice);
```



# LONG PARAMETER LIST - TREATMENT 2

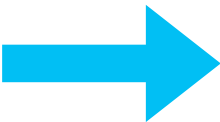
```
double finalPrice =  
    discountedPrice(basePrice);
```

Replace Parameter with Method Call

Customer
amountInvoicedIn (date : DateRange) amountReceivedIn (date : DateRange) amountOverdueIn (date : DateRange)

Introduce Parameter Object

Customer
amountInvoicedIn (start : Date, end : Date) amountReceivedIn (start : Date, end : Date) amountOverdueIn (start : Date, end : Date)



Customer
amountInvoicedIn (date : DateRange) amountReceivedIn (date : DateRange) amountOverdueIn (date : DateRange)

# LONG PARAMETER LIST - TREATMENT 3

```
double finalPrice =  
    discountedPrice(basePrice);
```

Replace Parameter with Method Call

Customer
amountInvoicedIn (date : DateRange) amountReceivedIn (date : DateRange) amountOverdueIn (date : DateRange)

Introduce Parameter Object

```
plan.withinRange(daysTempRange);
```

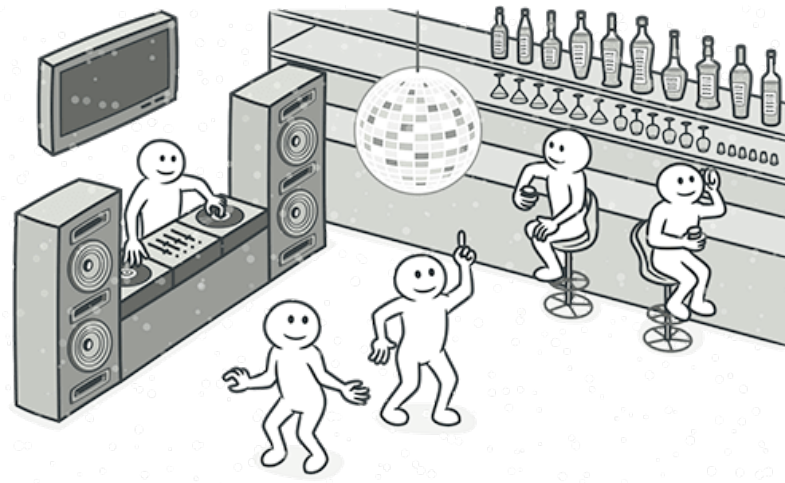
**Preserve Whole Object**

```
int low = daysTempRange.getLow();  
int high = daysTempRange.getHigh();  
boolean withinPlan = plan.withinRange(low, high);
```



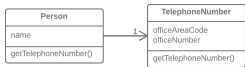
```
boolean withinPlan =  
    plan.withinRange(daysTempRange);
```

# DATA CLUMPS



Identical groups of variables

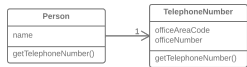
# DATA CLUMPS - TREATMENT 1



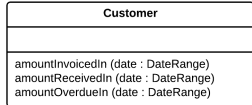
## Extract Class



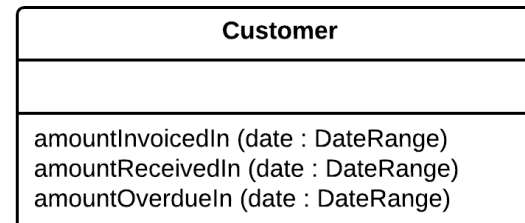
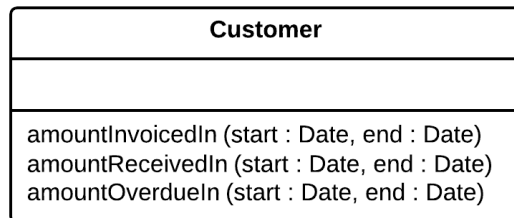
# DATA CLUMPS - TREATMENT 2



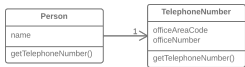
## Extract Class



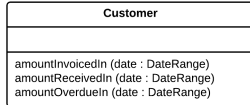
## Introduce Parameter Object



# DATA CLUMPS - TREATMENT 3



Extract Class



Introduce Parameter Object

```
plan.withinRange(daysTempRange);
```

Preserve Whole Object

```
int low = daysTempRange.getLow();
int high = daysTempRange.getHigh();
boolean withinPlan = plan.withinRange(low, high);
```



```
boolean withinPlan =
    plan.withinRange(daysTempRange);
```



# Questions?



[marco.grueter@css.ch](mailto:marco.grueter@css.ch)



<https://sourcemaking.com/>  
<https://refactoring.guru/>