

Hello



**MOVEMBER IS THE LEADING CHARITY CHANGING THE FACE OF MEN'S HEALTH
ON A GLOBAL SCALE,
FOCUSING ON MENTAL HEALTH AND SUICIDE PREVENTION, PROSTATE CANCER
AND TESTICULAR CANCER.**





RECIPES FOR WHAT YOU RECEIVE
AND WHAT YOU RETURN



Arguments and Results

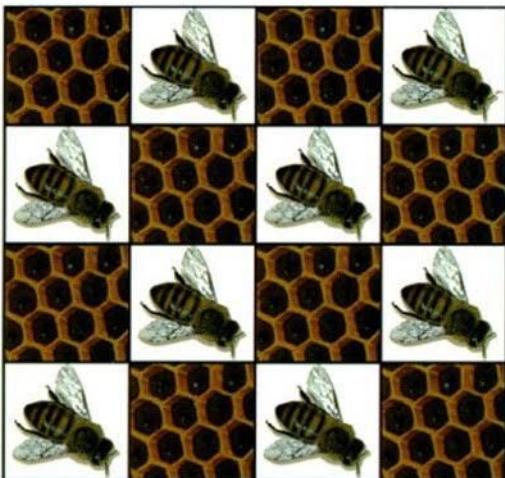
James Noble

MRI, School of MPCE,
Macquarie University, Sydney.

kjx@mri.mq.edu.au

October 20, 1997

SMALLTALK BEST PRACTICE PATTERNS

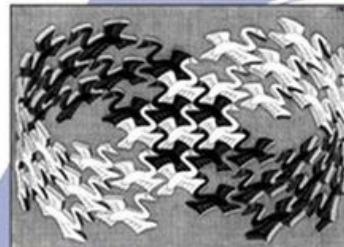


KENT BECK

Design Patterns

Elements of Reusable
Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch



What you receive



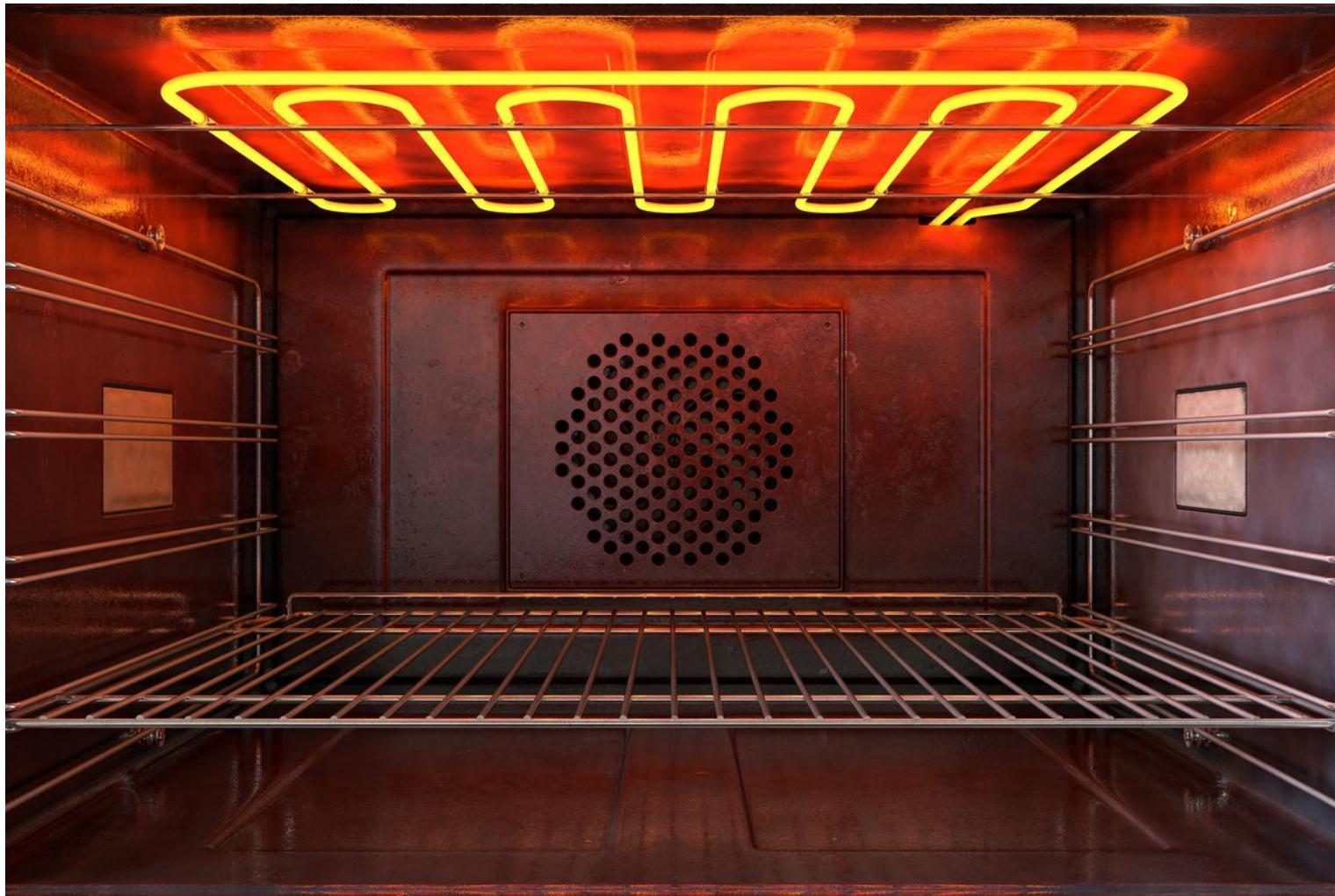
7009



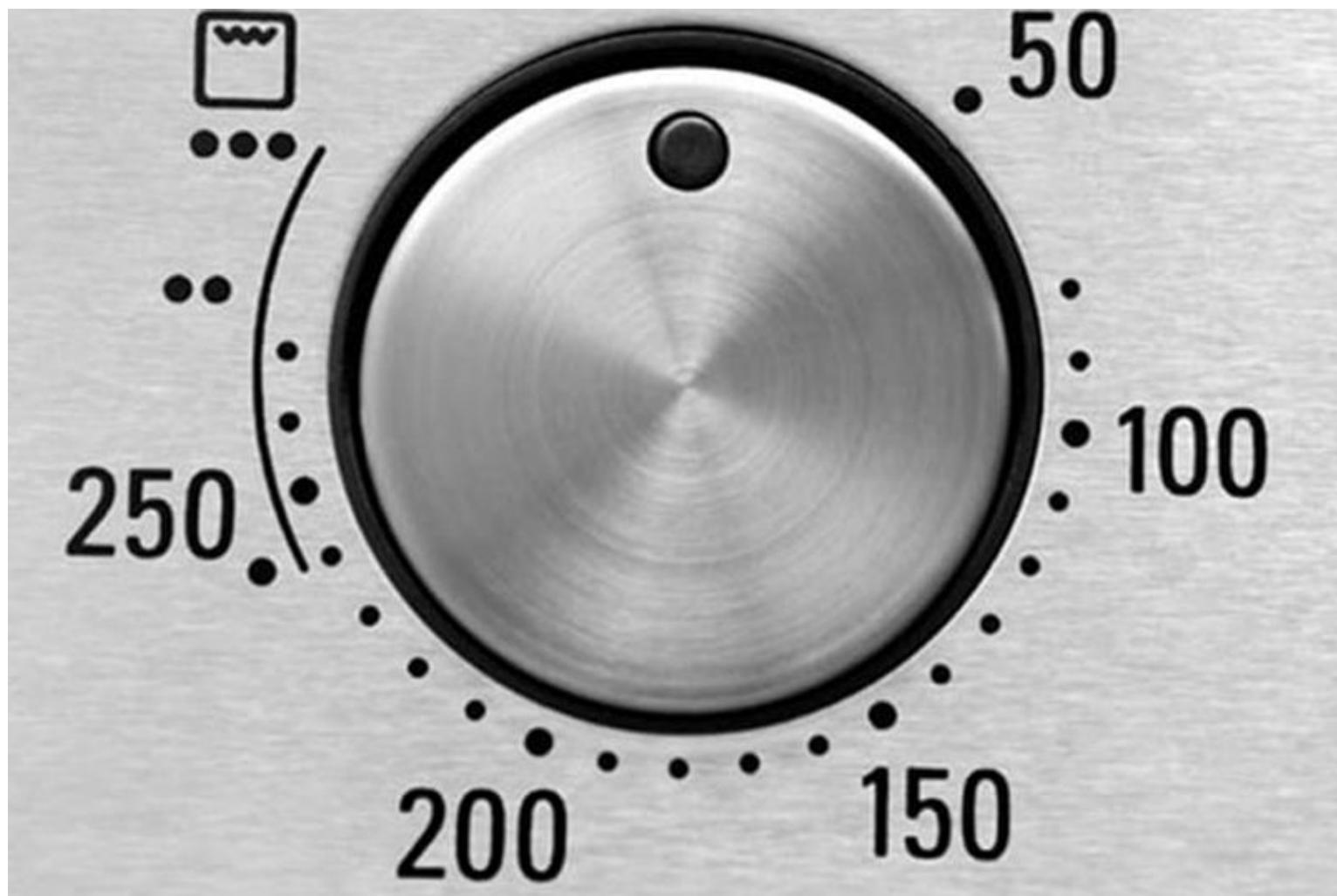




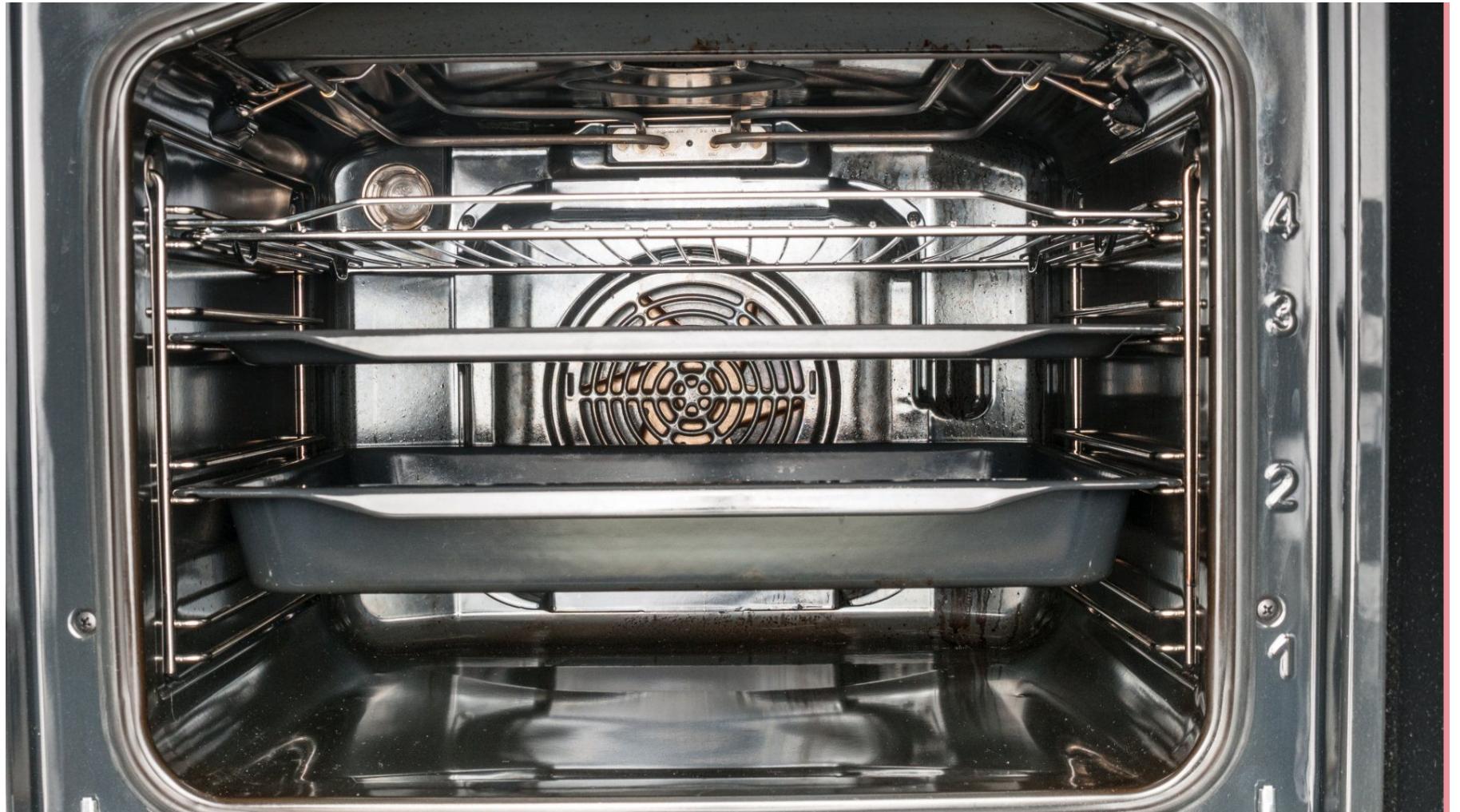








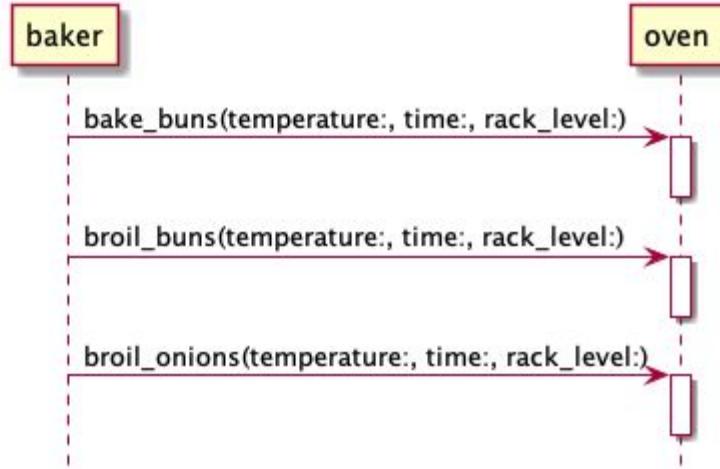




Arguments Object

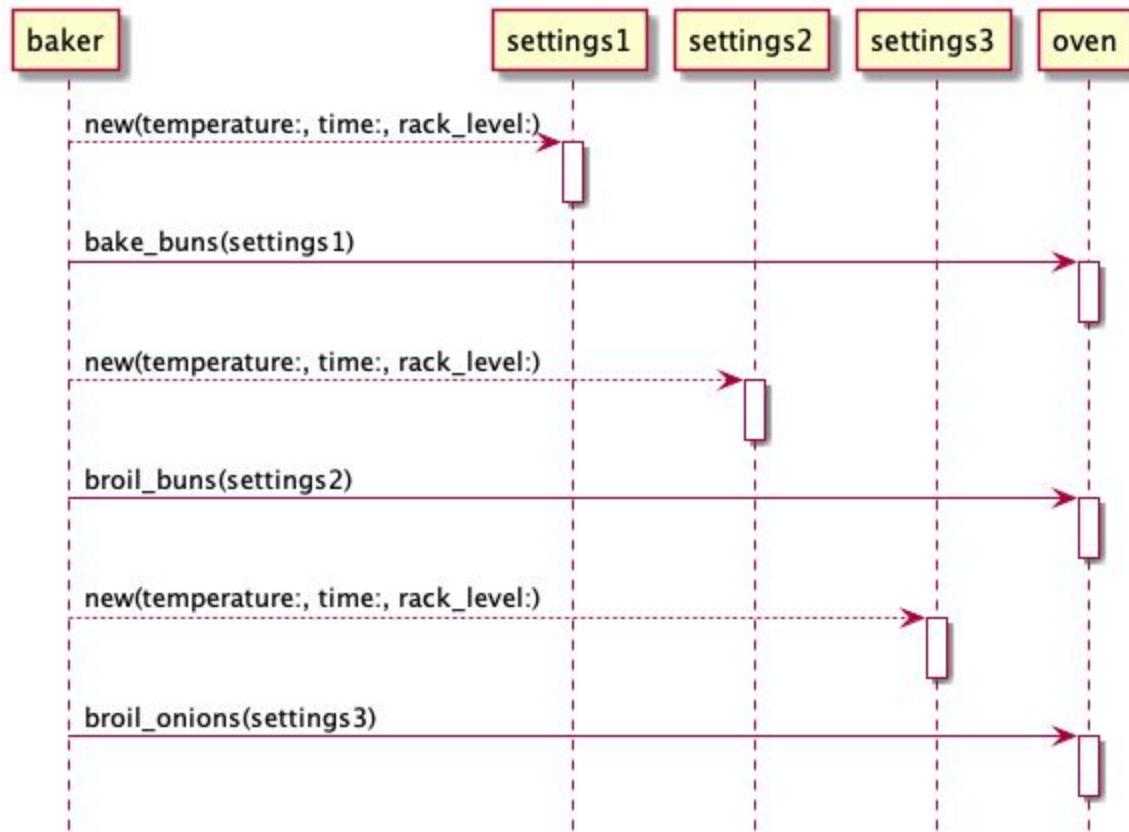


```
1 class Oven
2   def bake_buns(temperature:, time:, rack_level:); end
3
4   def broil_buns(temperature:, time:, rack_level:); end
5
6   def broil_onions(temperature:, time:, rack_level:); end
7 end
```





```
1 settings = OvenSettings.new(temperature: 200, time: 5, rack_level: :top)
2
3 oven.bake_buns(settings)
```



Selector Object



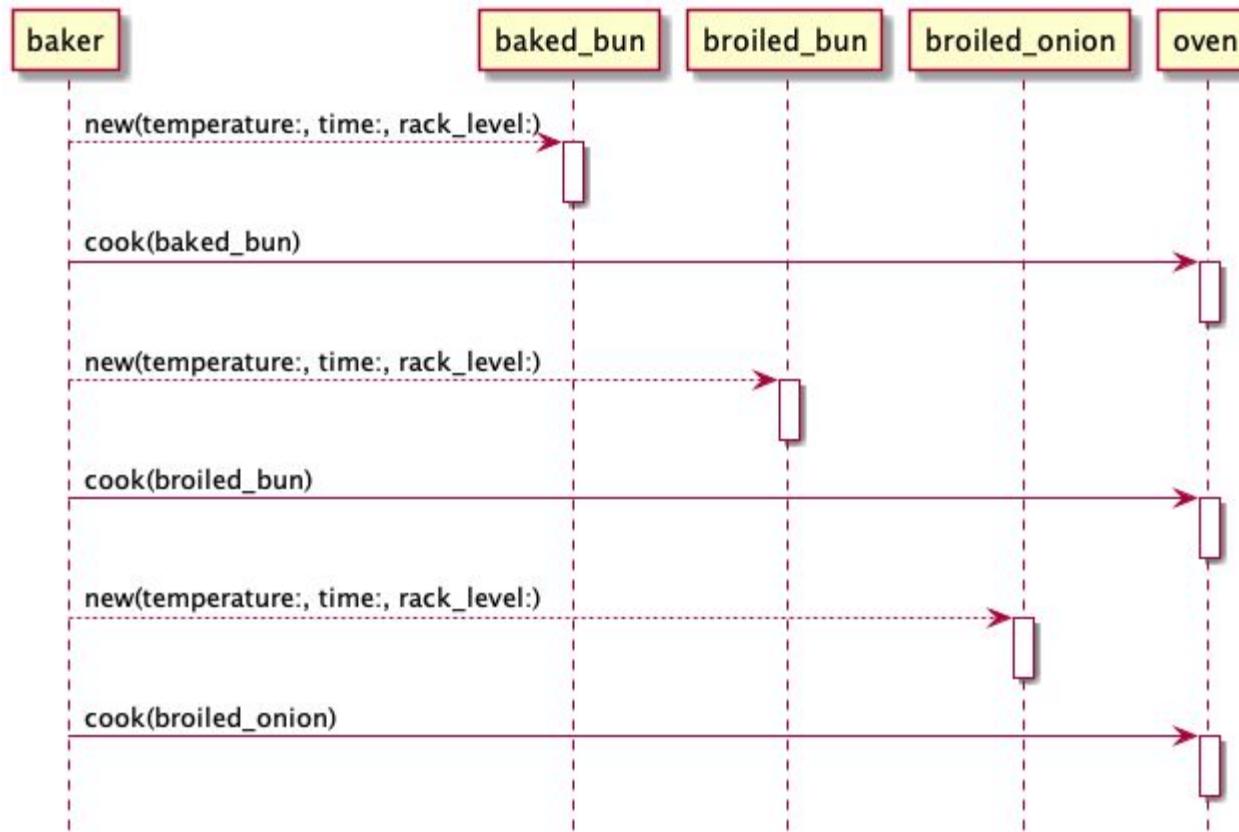
```
1 oven.bake_buns(settings)
2 oven.broil_buns(settings)
3 oven.broil_onions(settings)
```



```
1 recipe = OvenRecipe.new(type: :baked_bun, temperature: 200, time: 5, rack_level: :top)
2
3 oven.cook(recipe)
```



```
1 recipe = BakedBunRecipe.new(temperature: 200, time: 5, rack_level: :top)
2
3 oven.cook(recipe)
```



Curried Object

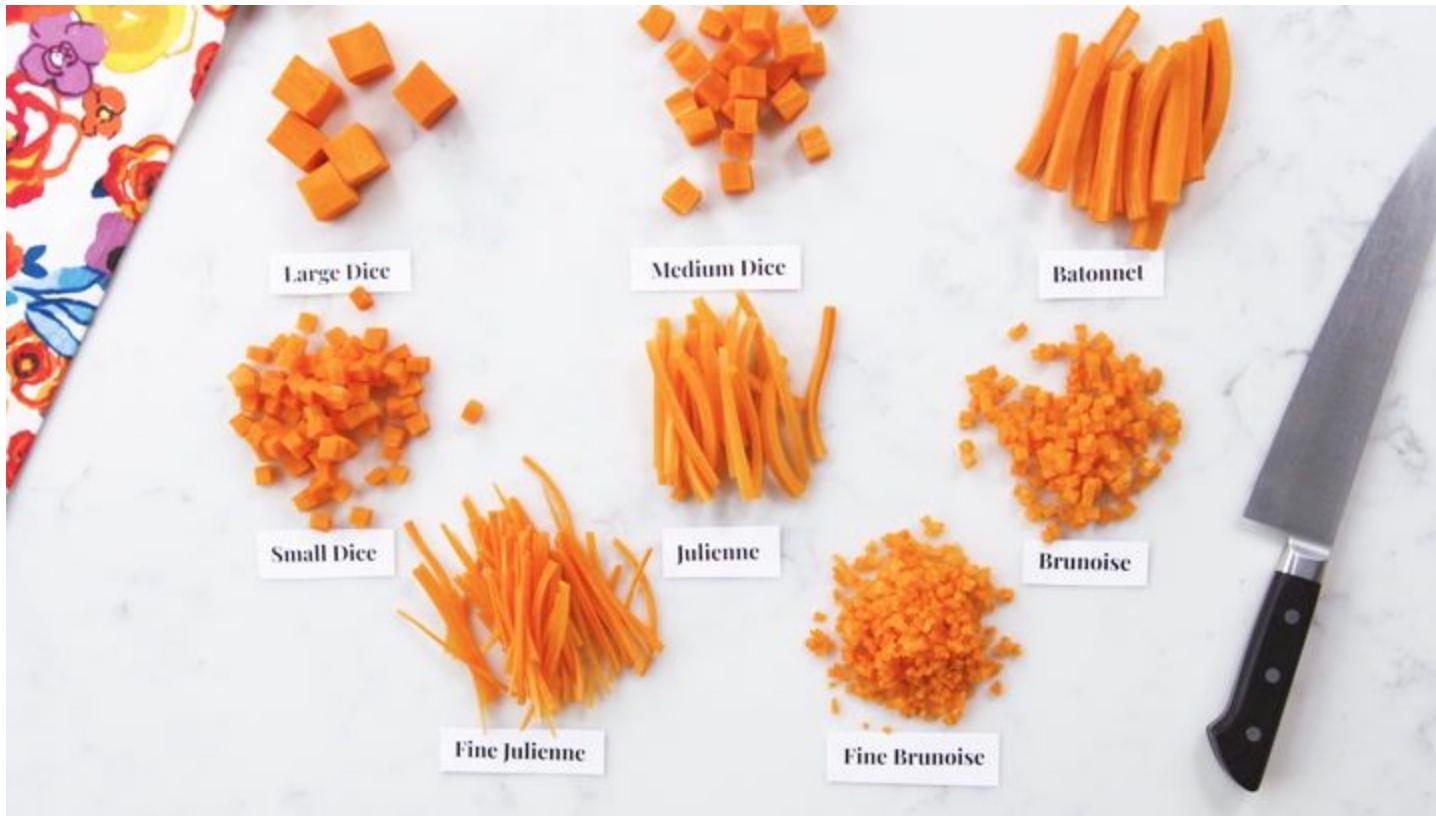






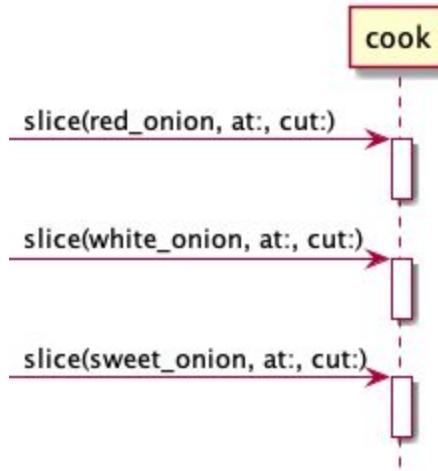
Julienne, allumette, or french cut





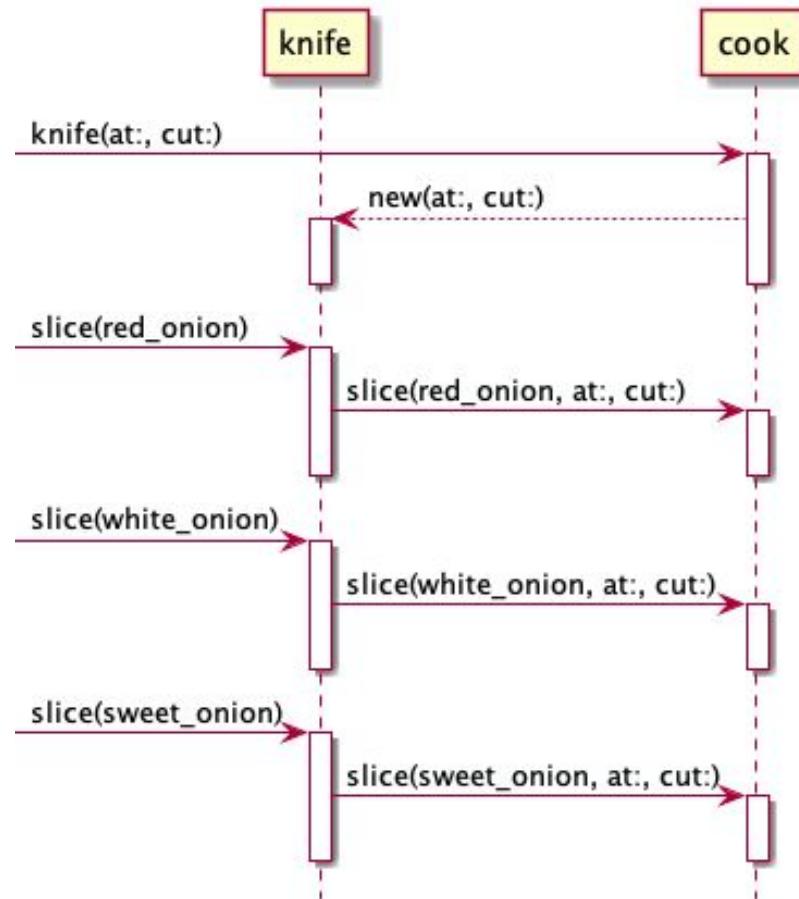


```
1 cut = Cut.new(type: :julienne)
2 offset = Distance.new(0, cut.width)
3
4 cook.slice(red_onion, at: origin, cut: cut)
5 cook.slice(white_onion, at: (origin + offset), cut: cut)
6 cook.slice(sweet_onion, at: (origin + offset * 2), cut: cut)
```





```
1 knife = cook.knife(at: origin, cut: Cut.new(type: :julienne))
2
3 knife.slice(red_onion)
4 knife.slice(white_onion)
5 knife.slice(sweet_onion)
```



What you return





175524745



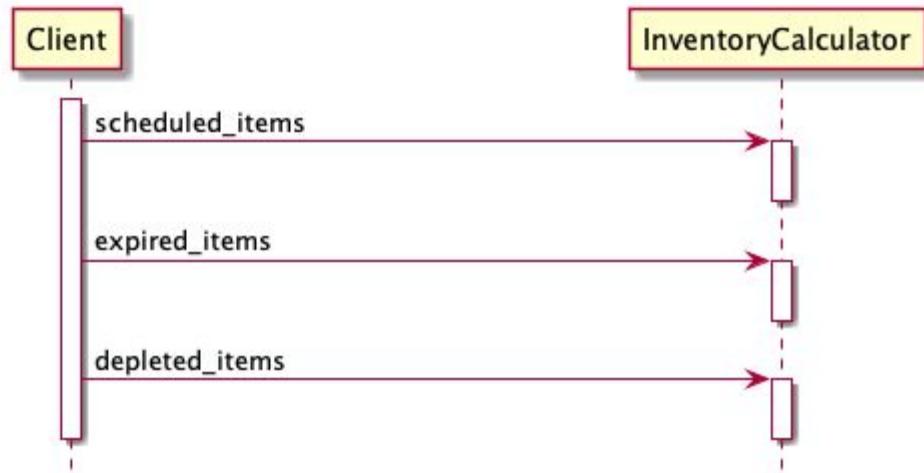
Result Object



```
1 calc = InventoryCalculator.new(stock)
2
3 calc.depleted_items
4 calc.expired_items
5 calc.scheduled_items
```



```
1 class InventoryCalculator
2   # ...
3
4   def scheduled_items
5     @scheduler_client.fetch_items(since: beginning_of_month)
6   end
7 end
```





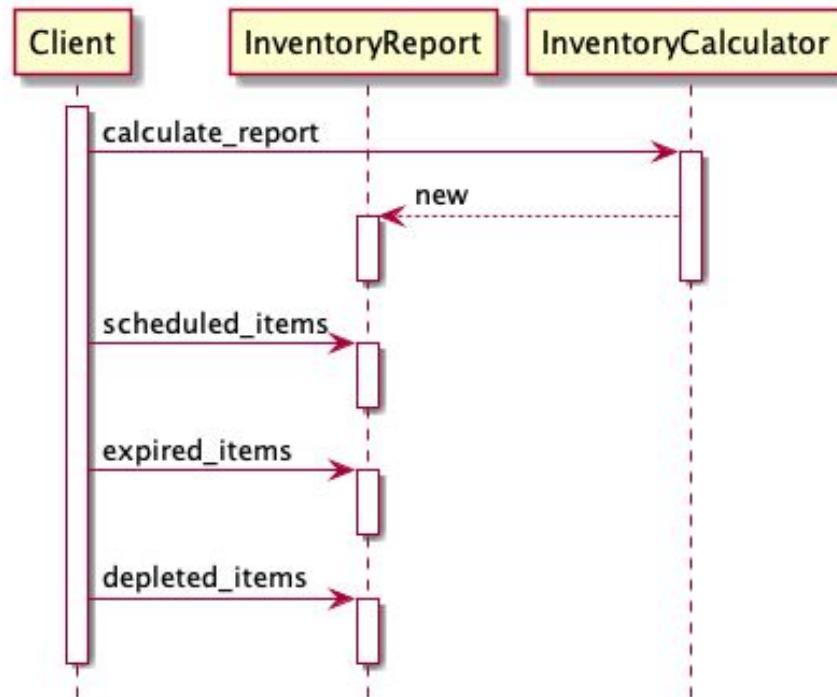
```
1 class InventoryReport
2   attr_reader :expired_items, :scheduled_items, :depleted_items
3
4   def initialize(expired_items:, scheduled_items:, depleted_items:)
5     @expired_items = expired_items
6     @scheduled_items = scheduled_items
7     @depleted_items = depleted_items
8   end
9 end
```



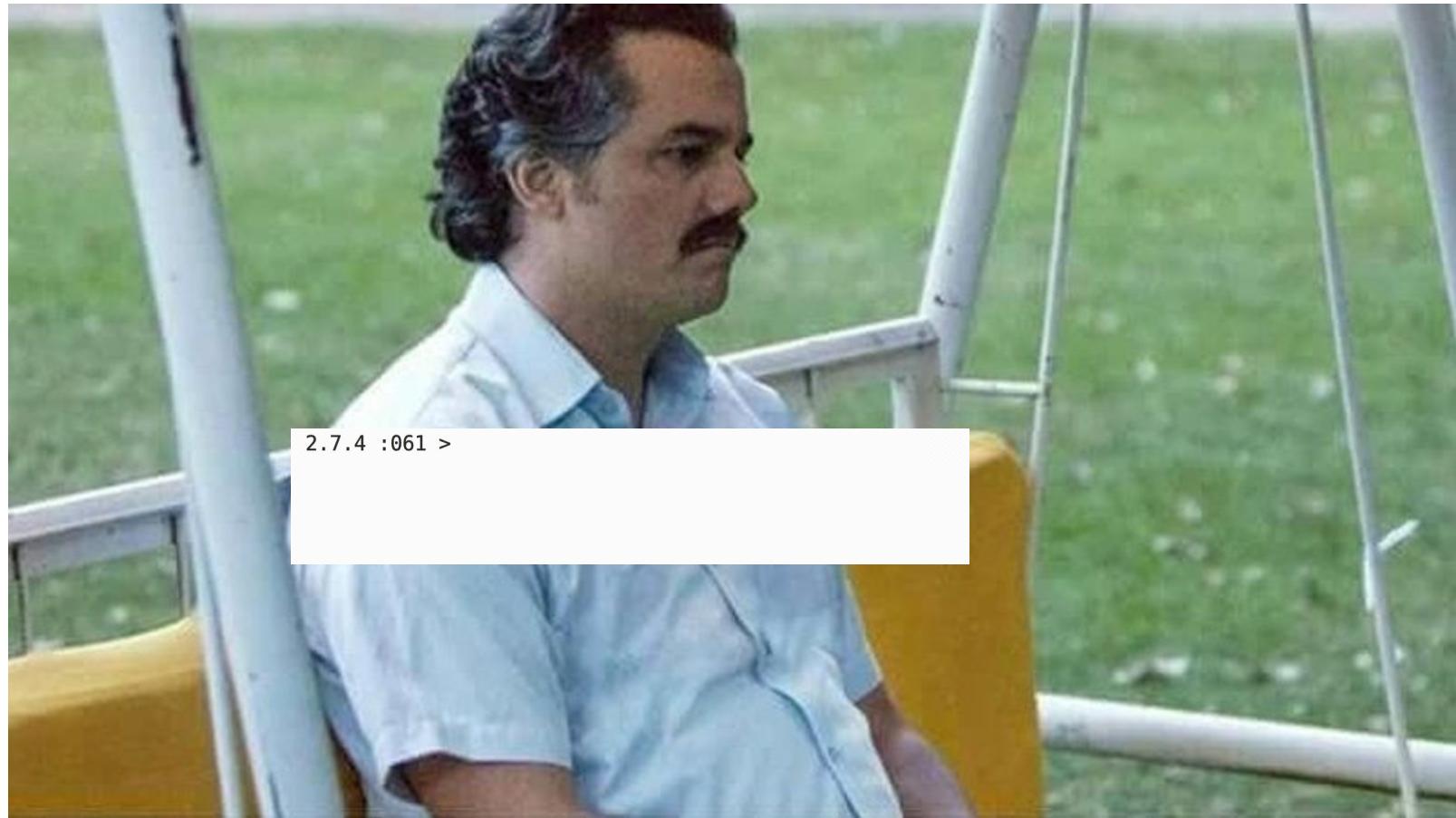
```
1 class InventoryCalculator
2   # ...
3
4   def calculate_report
5     scheduled_items = @scheduler_client.fetch_items(since: beginning_of_month)
6     InventoryReport.new(scheduled_items: scheduled_items)
7   end
8 end
```



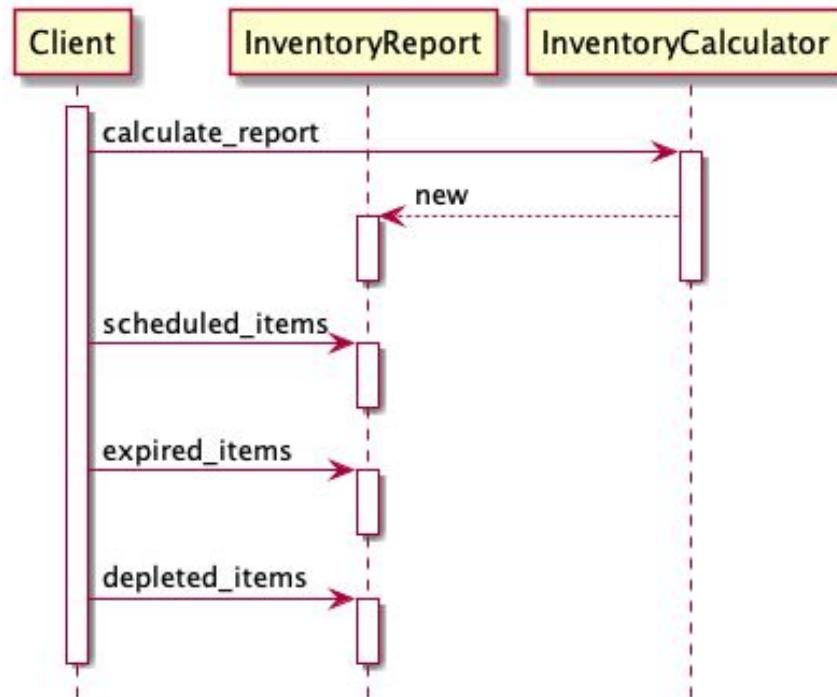
```
1 inventory_calculator = InventoryCalculator.new(stock)
2 inventory_report = inventory_calculator.calculate_report
3
4 inventory_report.expired_items
5 inventory_report.scheduled_items
6 inventory_report.depleted_items
```



Future Object



2.7.4 :061 >





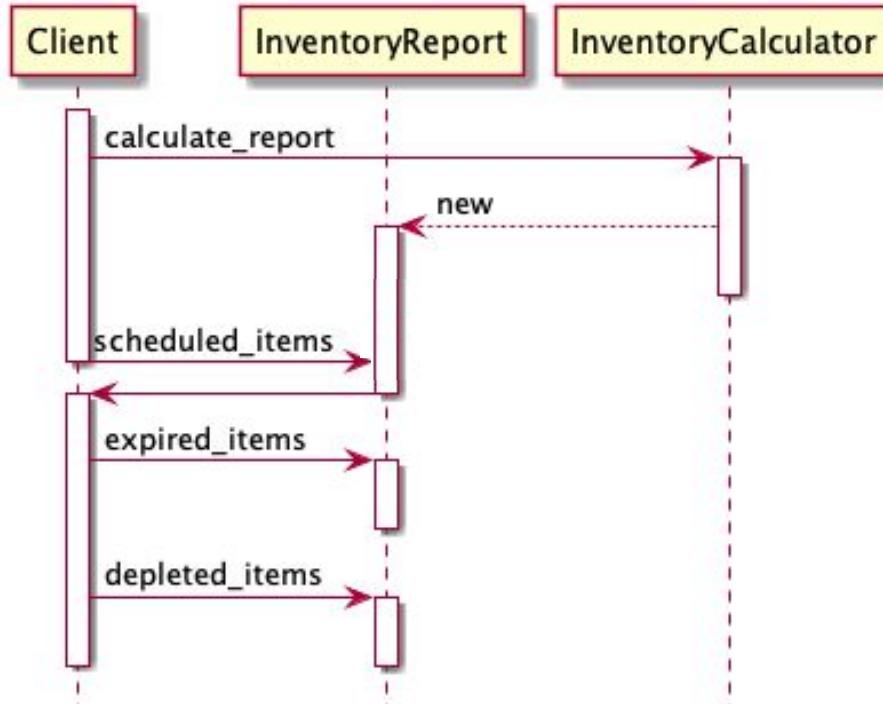
```
1 class Future < Delegator
2   def initialize(&block)
3     @block = block
4     @thread = Thread.new { @block.call }
5   end
6
7   def __getobj__
8     @thread.value
9   end
10 end
```



```
1 class InventoryCalculator
2   # ...
3
4   def calculate_report
5     scheduled_items = -> { @scheduler_client.fetch_items(since: beginning_of_month) }
6     InventoryReport.new(scheduled_items: scheduled_items)
7   end
8 end
```

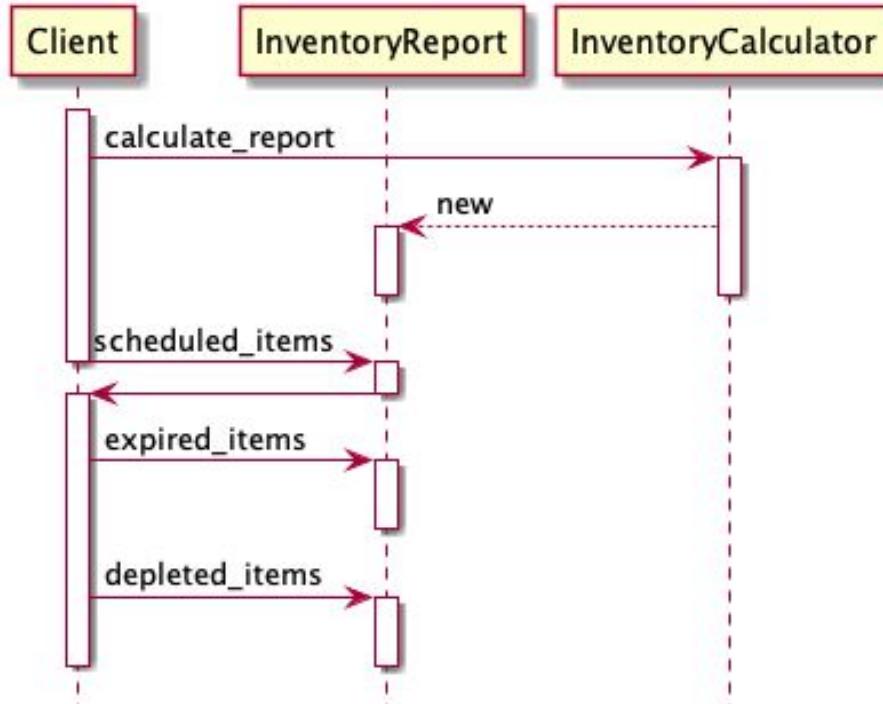


```
1 class InventoryReport
2   attr_reader :scheduled_items
3
4   def initialize(scheduled_items:)
5     @scheduled_items = Future.new(&scheduled_items)
6   end
7 end
```



Lazy Object







```
1 class InventoryCalculator
2   # ...
3
4   def calculate_report
5     scheduled_items = -> { @scheduler_client.fetch_items(since: beginning_of_month) }
6     InventoryReport.new(scheduled_items: scheduled_items)
7   end
8 end
```



```
1 class InventoryReport
2   attr_reader :scheduled_items
3
4   def initialize(scheduled_items:)
5     @scheduled_items_proc = scheduled_items
6   end
7
8   def scheduled_items
9     @scheduled_items ||= @scheduled_items_proc.call
10  end
11 end
```



IN-N-OUT
HAMBURGERS
NO DELAY

CHEESEBURGER 50
HAMBURGER 45
FRENCH FRIES 20
PIPS 10
MILK 10
COFFEE 10
SODA 10
CHOCOLATE 10

DOUBLE-DOUBLE

IN-N-OUT
BURGER



Thank you

Walerian Sobczak