

Lecture 9

Wednesday Oct. 11

Iterator Design Pattern

CLIENT

CLIENT APPLICATION+

```

contains: ITERABLE+
  -- Fresh cursor of the container.
increase_balance(v: INTEGER; name: STRING)
  -- Increase the balance for account with owner name.
? across container as cur
  all
    cur.item.balance ≥ v
  end
! across old container.deep_twin as cur
  all
    (cur.item.owner ~ name implies
     cur.item.balance = old cur.item.balance + v)
  and
    (cur.item.owner ~ name implies
     cur.item.balance = old cur.item.balance)
  end
some_account_negative: BOOLEAN
  -- Is there some account negative?
! Result =
  across container as cur
  some
    cur.item.balance < v
  end
  
```

program against interface (ITERABLE) (ARRAY, LIST)

SUPPLIER

ITERABLE *

```

new_cursor*: ITERATION_CURSOR[G]
  -- Fresh cursor associated with current structure.
Result: T
  
```

container+

new_cursor*

ITERATION_CURSOR[G] *

```

after*: BOOLEAN
  -- Are there no more items to iterate over?
item*: G
  -- Item at current cursor position.
? valid_position: not after
forth*
  -- Move to next position.
? valid_position: not after
  
```

Library classes (iteration cursor already imp.)

ARRAY[G] +

new_cursor+

INDEXABLE_ITERATION_CURSOR[G] +

```

after+: BOOLEAN
  -- Are there no more items to iterate over?
item+: G
  -- Item at current cursor position.
forth+
  -- Move to next position.
start+
  -- Move to first position.
  
```

LINKED_LIST[G] +

ARRAYED_LIST[G] +

ITERABLE COLLECTION

deferred class ≈ abstract class

lucky

new_cursor

ITERATION_CURSOR

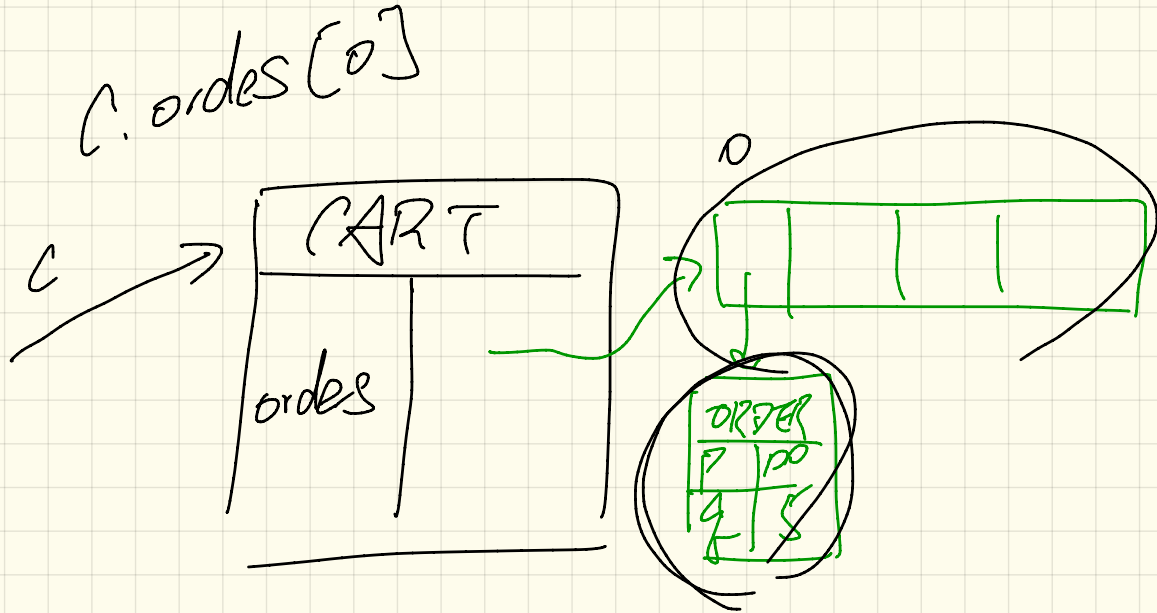
after item forth start

case where the iteration-cursor is not directly supported.

CLASS CART inherit ITERABLE (ORDER)

orders: ARRAY (ORDER)

end



+

ARIZAL

*

ITERABLE-

+

effective

concrete/non-abstract.

*

deferred

abstract

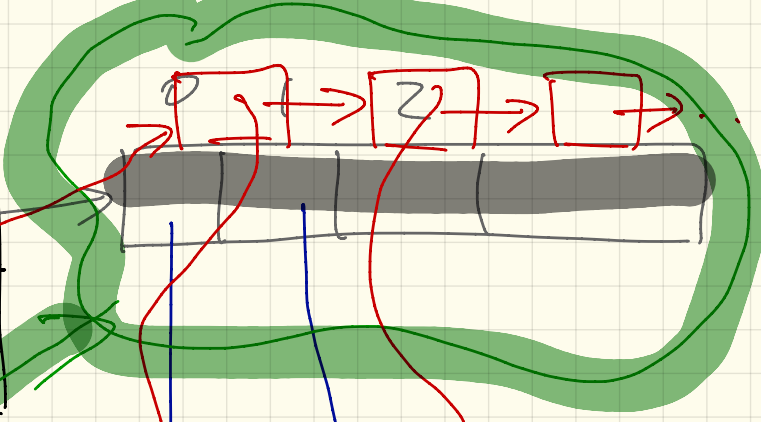
implementation secrets (should be hidden)

- data structure (A, L)
- algorithm.

change on supplier

ARRAY → linked list

CART	
orders	
new_supplier	



CART
THIS IS THE
ONCE WAY
FOR CLIENTS
TO ACCESS
THE CART
ORDERS.

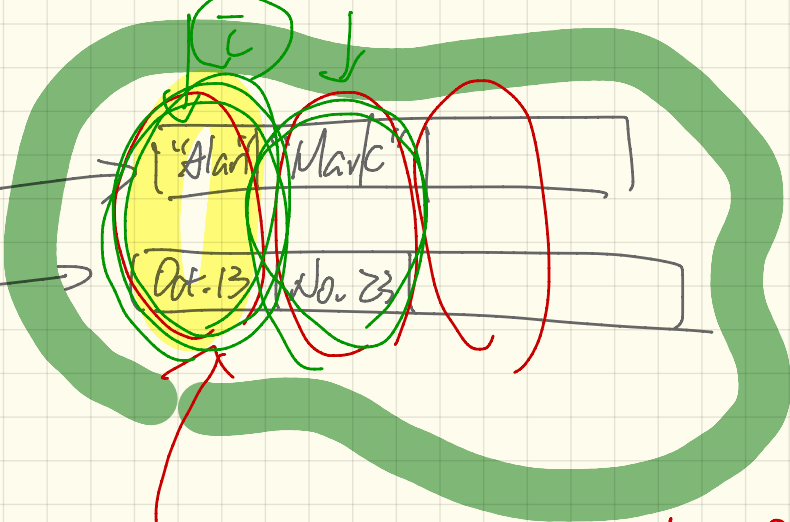
ITERATIVE APPROACH
FROM
after
forth

ORDER	
P	20
Q	4

ORDER	
P	40
Q	3

bb →

Book	
names	
records	
New-Cursor	



MY_ITERATION_CURSOR

ITERATION_CURSOR
item
index
offset
(i): INTEGER I

each time during
 the iteration
 we retrieve a
 (name, record) pair
 ↳ 1. TUPLE
 2. PAIR class