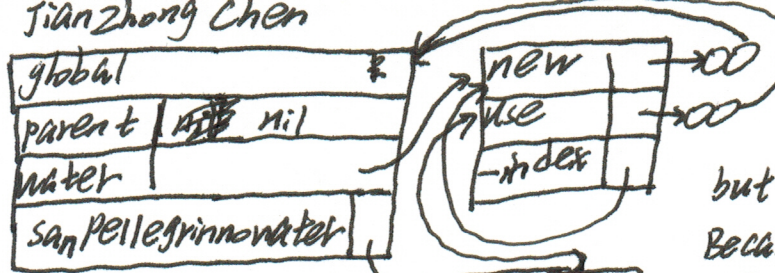
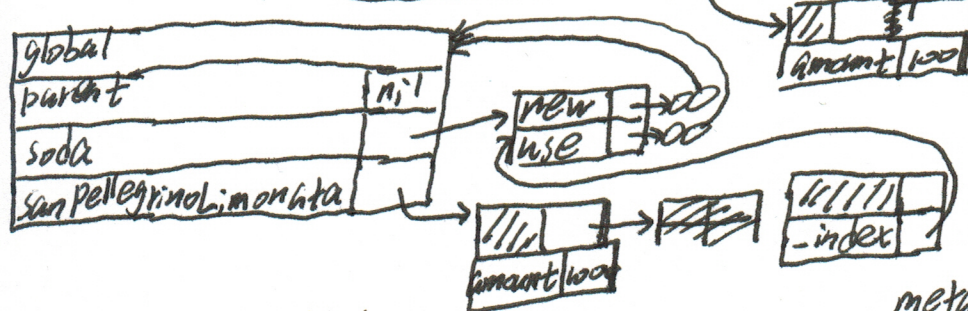


1. a)  $>90$  b)  $>90$



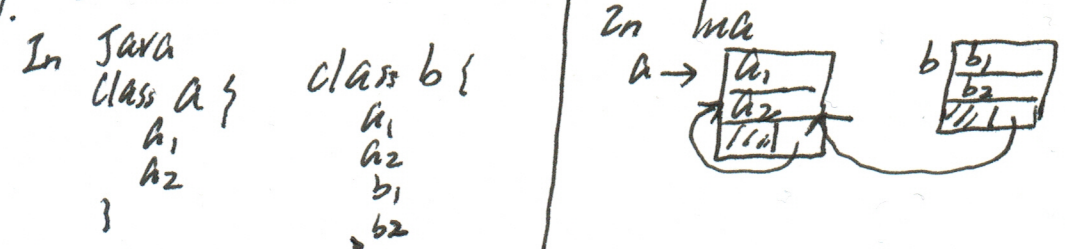
c) sanpellegrinolimonata creates a new metatable but sanpellegrinowater does not.



Because there is an extra metatable in the ~~Re~~inheritance of sanpellegrinolimonata, there is an extra reference. The references of the object are not affected. The usage of metatable and -index save ~~the~~ space.

d)  $spw = sanpellegrinowater.new(\{amount = 1000\})$   
 e) Java or Python uses class based inheritance. class base inheritance save time, prototype base inheritance save space.

eg. b inheritant a, a has field  $a_1, a_2$  and b has field  $b_1, b_2$



2. a)  $HBox.children[i].y = HBox.y$   
 $HBox.height = \max(HBox.children.map(\lambda c \{c.height\}))$   
 $VBox.children[i].x = VBox.x$   
 $VBox.children[i].y = VBox.y + \sum(VBox.children[0 \text{ to } i-1].map(\lambda c \{c.height\}))$   
 $VBox.width = \max(VBox.children.map(\lambda c \{c.width\}))$   
 $VBox.height = \sum(VBox.children.map(\lambda c \{c.height\}))$   
 b) For each node, 2 pass is needed to compute 4 attributes.

c) Pass #

|   |           |  |
|---|-----------|--|
| 1 | bottom up | $HBox.x = HBox.x + \sum(HBox.children[0:i-1].map(\lambda c \{c.width\}))$<br>$HBox.y = HBox.children[i].y$                       |
| 2 | bottom up | $HBox.width = \sum(HBox.children.map(\lambda c \{c.width\}))$<br>$HBox.height = \max(HBox.children.map(\lambda c \{c.height\}))$ |

3. a) ①  $x$  can't compare to int  
 ② after we  $y = null$ ,  $y \times 4$  will raise an error.

b) function main(x: int): int {  
 def y: int = 1  
 if (x < 10) {  
 x(1)  
 } else {  
 y = null  
 }  
 y \* 4  
}

c) if y = null, y \* 4 will still raise an error  
 x(1) will still raise function not defined

d) function main(x) {  
 if (x > 10) {  
 return 1  
 } else {  
 return "wow"  
 }  
}

e) even though n is a nonZeroInt when initialized, but the value could still change to zero during run time.

eg n: nonZeroInt = 1      n + x = 0.  
 x: nonZeroInt = -1

f) string[] a = new String[1];  
 object[] b = a;  
 b[0] = new Integer(0);

a[0].toUpperCase(); → will lead to error.

In Java, when the program executes to line 3. Java will prevent assign b[0] to be a new int, because ~~the~~ b[0] = a[0] which defined as a string type and can't assigned to an int.