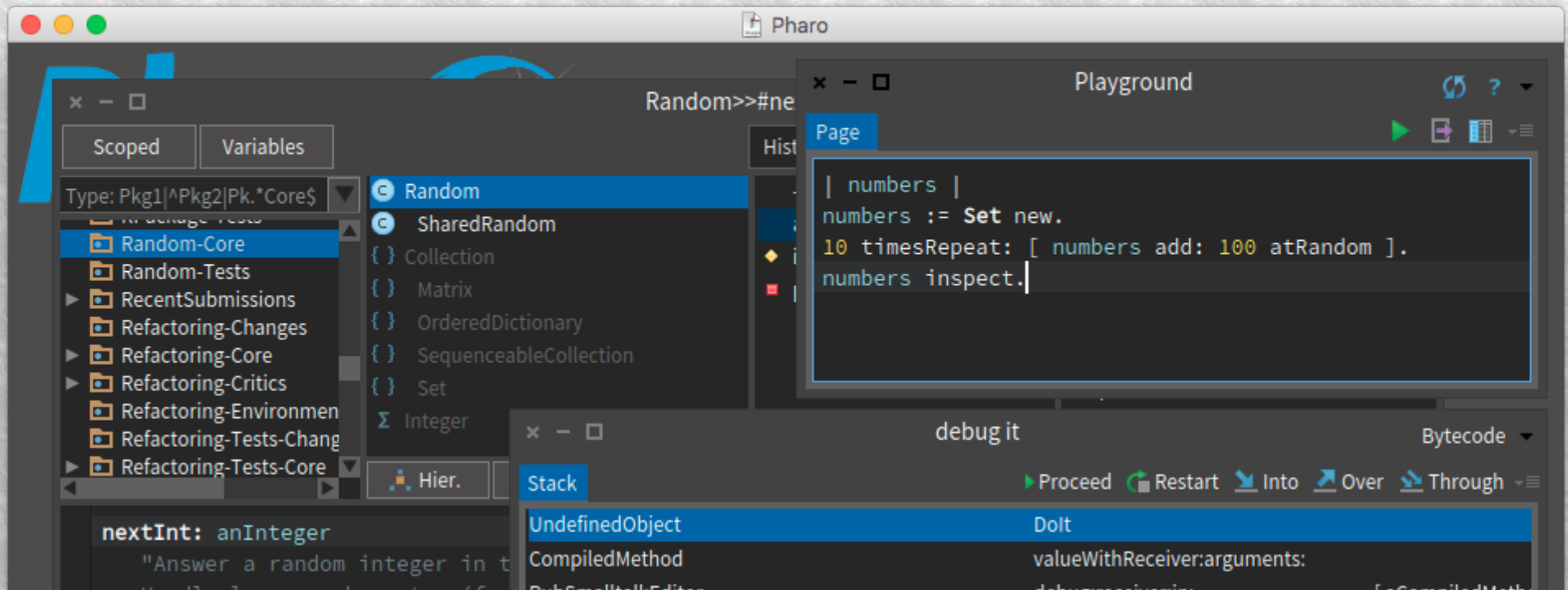


LA SYNTAXE

Marie Gallardo
Gaultier Parain
Imrane Bensadok

INTRODUCTION

- Langage orienté objet
- Tout est objet, message ou bloc



SOMMAIRE

I. Pharo en général

- Éléments de syntaxes
- Variables
- Les types de messages
- Ordre des messages et opérations
- Les blocks

II. Comment passer de la syntaxe Java à celle de Pharo ?

- Les boucles de base
- Les cascades , self, super
- Définitions des méthodes et classes

III. Les erreurs de syntaxe courantes en Pharo

ÉLÉMENTS DE SYNTAXE

Commentaire	« commentaire »
Caractère	\$c \$# \$@
Chaîne de caractères	'toto' 'l'"idiot'
Symbole	#mac #+
Tableau littéral	#(24 5 111)
Entier	1, 2r101
Réel	1.5 6.03e-34,4, 2.4e7
Booléen	true, false (instances de True et False)
Indéfini	Nil (instance de UndefinedObject)
Point	25@150

VARIABLES

- Déclaration de variable temporaire :| var |
- Assignment de variable : var := aValue
- Séparateur : message . message
- Return : ^ expression

MESSAGES

	UNARY MESSAGE	BINARY MESSAGE	KEYWORD MESSAGE
Syntaxe	Receveur Sélecteur	Receveur Sélecteur Argument	Receveur Key1:Arg1 Key2:Arg2
Exemple(s)	2 squared. Date today.	1+2. 3@4.	2 between: 10 and: 20.

ORDRE DES MESSAGES

Messages entre parenthèses



Unary



Binary



Keywords



Puis de gauche à droite

BLOCS

`fct(x) = x*x+3`

`fct:=[:x|x*x+3]`

`fct value : 2`

`#(1 2 -4 -86)`

`do: [:each | Transcript show: each abs printString ; cr]`

`> 1`

`> 2`

`> 4`

`> 86`

FOR / WHILE

For (deux méthodes) :

```
4 timesRepeat: [ self doSomething ]  
0 to: 100 by: 3 do:
```

While :

```
[ ... ] whileTrue: [ ... ]
```

ITÉRATEURS

Itérateur	Exemple
do	<code> #(16 11 68 19) do: [:each Transcript show: each ; cr]</code>
select	<code> #(16 11 68 19) select: [:i i odd] > #(11 19)</code>
reject	<code> #(16 11 68 19) reject: [:i i odd] > #(16 68)</code>

IF

IfTrue et IfFalse sont des messages envoyés aux booléens.

```
Weather isRaining
```

```
ifTrue: [ self takeMyUmbrella ]
```

```
ifFalse: [ self takeMySunglasses ]
```

CASCADE

Transcript

cr ;

show: 1 ;

show: 2

DEFINITION DES MÉTHODES

Les méthodes sont toutes publiques et retournent par défaut self

```
messageSelectorAndArgumentNames  
"comment stating purpose of message"  
| temporary variable names |  
statements
```

DÉFINITION DES CLASSES

Object subclass : #NomDeClasse

instancesBariablesNames :

classVariableNames :

category :

LISTES LITTERALE VS LISTES DYNAMIQUES

$A \rightarrow \#(1\ 2\ 6\ 1+1)$

$B \rightarrow \{1\ .\ 2+1\ .\ 6\ .\ 58\}$

LISTES LITTERALE VS LISTES DYNAMIQUES

$A \rightarrow \#(1\ 2\ 6\ 1+1) \longrightarrow (1\ 2\ 6\ 1\ \#+\ 1)$

$B \rightarrow \{1\ .\ 2+1\ .\ 6\ .\ 5*0\} \longrightarrow (1\ 3\ 6\ 0)$

TRUE VS TRUE

```
Book>>initialize  
InLibrary:=True
```

TRUE VS TRUE

Book>>initialize
InLibrary:=~~True~~ → true

PARENTHÈSES

$$3+2*10$$

$$1/3+2/3$$

PARENTHÈSES

$$3+2*10$$

$$>5*10$$

$$>50$$

$$1/3+2/3$$

$$>7/3/3$$

$$>7/9$$

SEPARATEURS

Attention à mettre des points comme
séparateurs et non des virgules !!!

DES QUESTIONS ?



Vu sur www.photos-a-la-con.fr