Les 1: De basiselementen

SESSIE 1

Les 1: De basiselementen

Programma's zijn bouwwerken. Je hebt basisbouwstenen nodig plus een aantal technieken om de basisbouwstenen te combineren tot grotere samengestelde elementen. Je hebt daarnaast dan nog technieken nodig deze combinaties te benoemen en te hergebruiken. Programmeren heeft niet zozeer te maken met weten welke basisbouwstenen in een gegeven programmeertaal voorhanden zijn maar veel meer met het kennen en kunnen toepassen van technieken om deze samen te stellen tot grote stabiele bouwwerken.

Elementen van een programmeertaal

- Elke min of meer fatsoenlijke programmeertaal kent :
  - **primitieve uitdrukkingen**: de bouwstenen
  - **combinatietechnieken**: de lijm om met de bouwstenen grotere constructies samen te stellen
  - **abstractietechnieken**: de techniek om de grotere constructies een naam te geven zodat ze daarna als een geheel kunnen gemanipuleerd en herbruikt worden
- In programmeertalen onderscheidt men twee groepen van objecten: **data-objecten** en **procedure-objecten**.
- In deze eerste les en een aantal volgende wordt alleen gebruik gemaakt van de primitieve data-bouwsteen 'number' [getal] en van basisoperaties over getallen [+,-,*, /]. Dit laat toe om de aandacht toe te spitsen op de combinatie- en abstractietechnieken voor procedure-objecten.

Elementen van een programmeertaal

	Data	Procedure
Primitive	5 5.0 -3 3.14	+ - * /
Combination		(* 3 5) (+ 1 2 3 4 5) (/ (* 3 5) (+ 1 2))
Abstraction	(define n 5)	(define (gemiddelde x y) (/ (+ x y) 2)

Terminologie (1)

getal	number
uitdrukking	expression
primitieve uitdrukking	primitive expression
primitieve operator	primitive operator
samengestelde uitdrukking	compound expression
vertolker - evaluator	interpreter
evaluatie	evaluation
samenstelling of combinatie	combination
operator - operanden	operator - operands
proceduretoepassing - oproep	procedure application - call
prefixnotatie	prefix notation
geneste uitdrukking	nested expression
variabele	variable
waarde	value

Terminologie (2)

omgeving	environment
globale omgeving	global environment
deeluitdrukking	subexpression
recursie	recursion
boom	tree
knopen	nodes
takken	branches
terminale knopen	terminal nodes
boom accumulatie	tree accumulation
speciale vorm	special form
proceduredefinitie	procedure definition
formele parameters	formal parameters
actuele parameters	actual parameters

De evaluator

(* 3 5)

15

De read-eval-print cyclus


```
> 5
5
> (* 3 5)
15
> (+ 1 2 3 4 5)
15
```

Uitdrukkingen

```
> 5
5
> -7
-7
> 3.14
3.14
> (* 3 5)
15
> (+ 1 2)
3
> (+ 1 2 3 4 5)
15
> (/ (+ 4 6) 2)
5
> (+ (* 2 5) (- 7 2))
15
> (- 6 (/ 12 4) (* 2 (+ 5 6)))
-19
```

primitieve uitdrukking

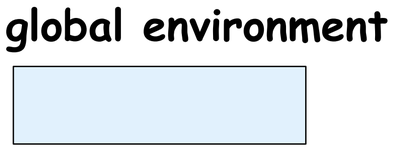
samengestelde uitdrukking

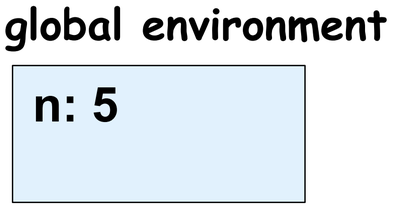
geneste samengestelde uitdrukking

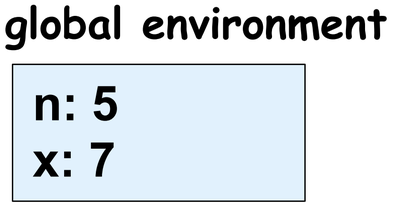
Variabelen (1)

```
> n
⊗⊗ n: undefined;
cannot reference undefined identifier
> (define n 5)
> n
5
> (* 4 n)
20

> (+ x 2)
⊗⊗ x: undefined;
cannot reference undefined identifier
> (define x 7)
> (+ x 2)
9
```





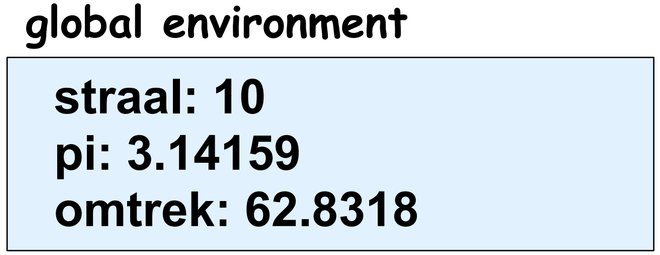


(define <name> <expression>)

special form

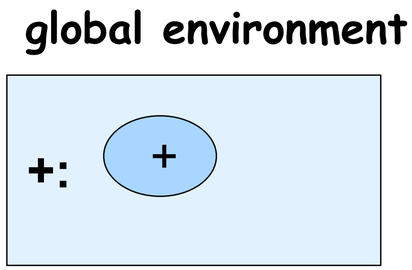
Variabelen (2)

```
> (define straal 10)
> (define pi 3.14159)
> (define omtrek (* 2 pi straal))
> omtrek
62.8318
```

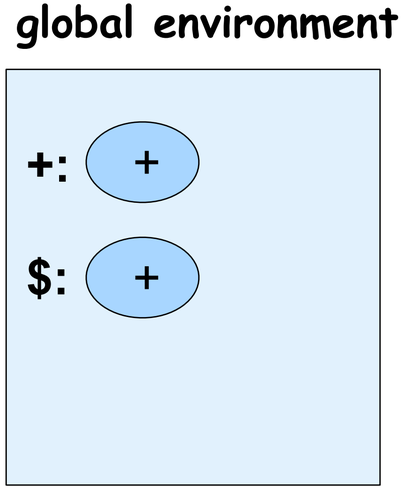


Variabelen (3)

```
> +
#<procedure:+>
> (+ 2 3)
5
```



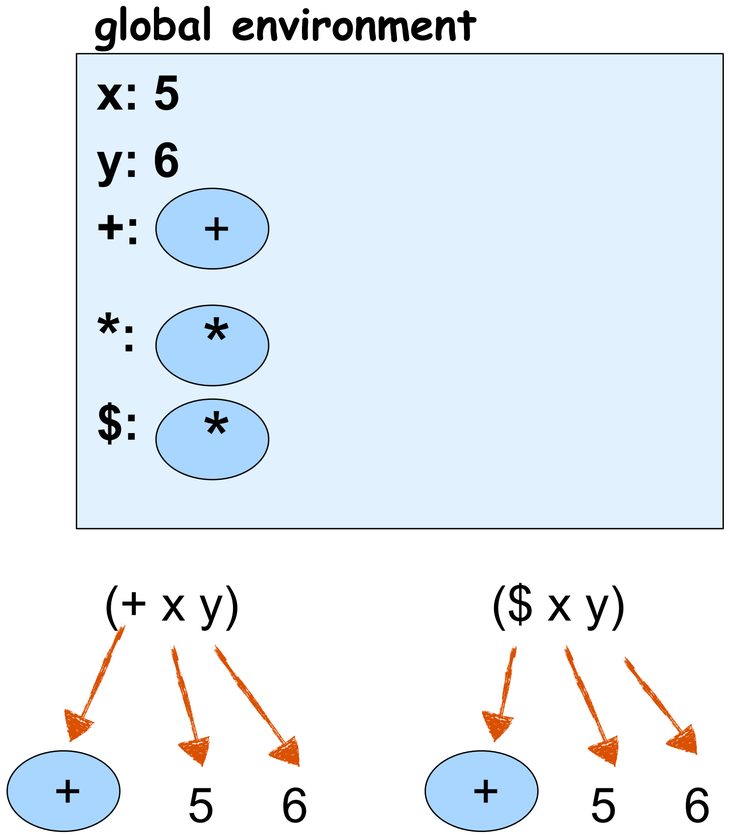

```
> $
⊗⊗ $: undefined;
cannot reference undefined identifier
> (define $ +)
> $
#<procedure:+>
> ($ 4 5)
9
```



Het omgevingsmodel voor evaluatie

```
> (define x 5)
> (define y 6)
> (+ x y)
11

> (define $ +)
> $
#<procedure:+>
> ($ x y)
11
```



Evaluatie: Samenvatting

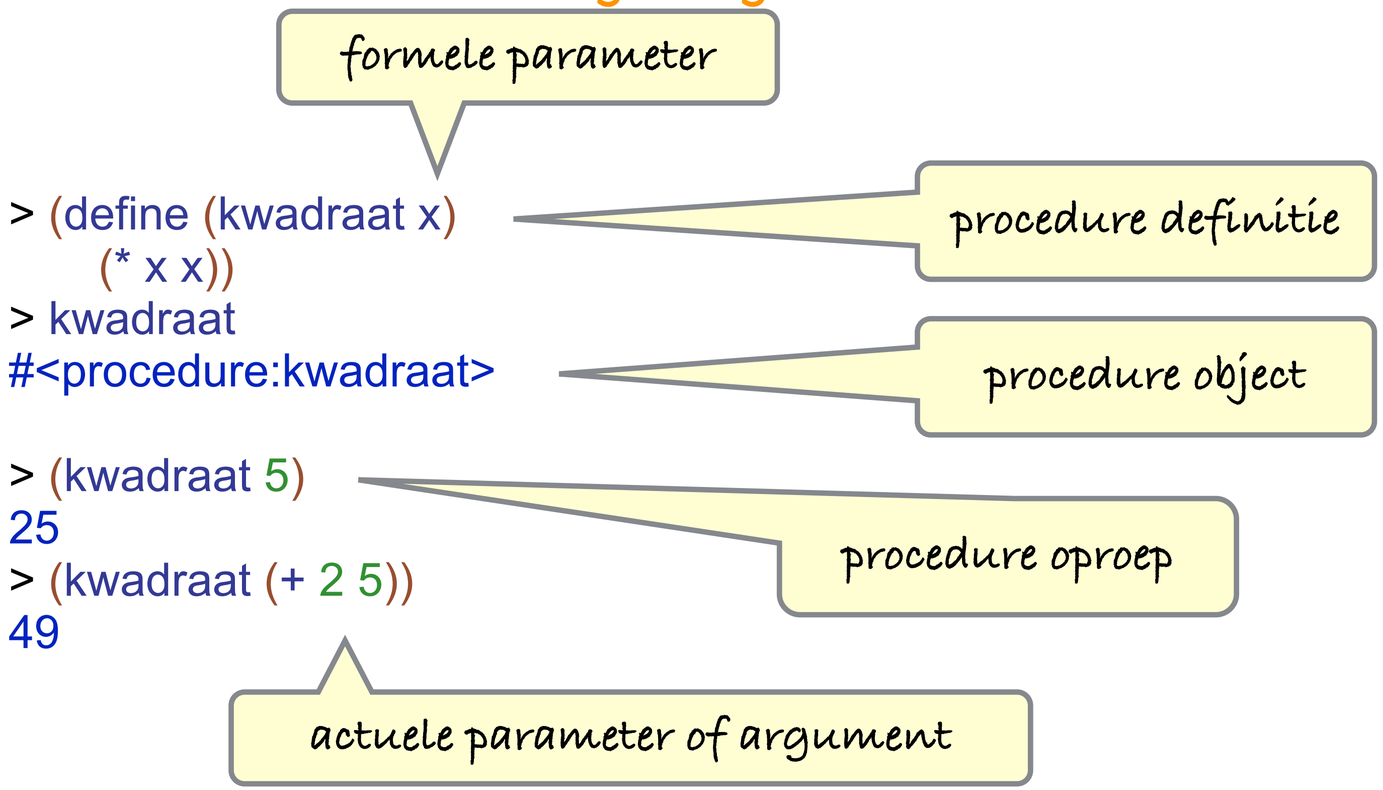
Number	▸	Number
Boolean	▸	Boolean
Symbol	▸	Object associated with symbol in environment
Combination	▸	Evaluate each subexpression and apply value of leftmost subexpression to the values of the other subexpressions
Special Form	▸	Dedicated behaviour

Les 1: De basiselementen

SESSIE 2

Procedurele abstractie

"om het kwadraat van 'iets' te berekenen,
vermenigvuldig 'iets' met zichzelf"



(define (<name> <formal-parameters>) <body>)

Modellen voor procedure evaluatie

The substitution model

(kwadraat 5)
(* x x)
(* 5 5)
25

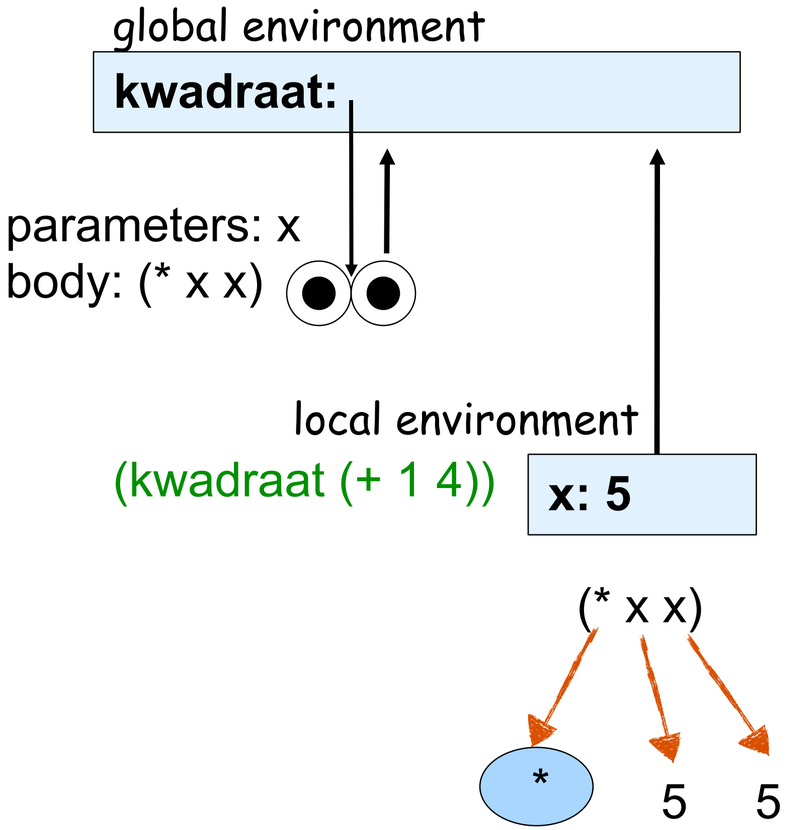
normal order

(kwadraat (+ 1 4))
(* x x)
(* (+ 1 4) (+ 1 4))
25

applicative order

(kwadraat (+ 1 4))
(* x x)
(* 5 5)
25

The environment model



Parameter binding is 'positioneel'

3 formele parameters in de proceduredefinitie

```
> (define (gemiddelde x y z)
    (/ (+ x y z) 3))
> (gemiddelde (* 2 3) (+ 1 2) (/ 6 2))
4
> (gemiddelde 3 4)
⊗⊗ gemiddelde: arity mismatch; the expected number of arguments
does not match the given number; expected: 3; given: 2;
arguments…: 3 4
> (gemiddelde 1 2 3 4)
⊗⊗ gemiddelde: arity mismatch; the expected number of arguments
does not match the given number; expected: 3; given: 4
arguments…:1 2 3 4
```

de volgorde van de argumenten bij de oproep is bepalend

er worden juist 3 argumenten verwacht

Uitgebreider voorbeeld

```
(define (kwadraat x)
 (* x x))

(define (som-kwadraten x y)
 (+ (kwadraat x) (kwadraat y)))

(define (f x)
 (som-kwadraten (+ x 1) (* x 2)))
```

$x \rightarrow x^2$

$x, y \rightarrow x^2 + y^2$

$x \rightarrow (x + 1)^2 + (x * 2)^2$

definities ingetikt in de editor

```
> (f 5)
136
```

interacties

Het substitutiemodel voor procedure evaluatie

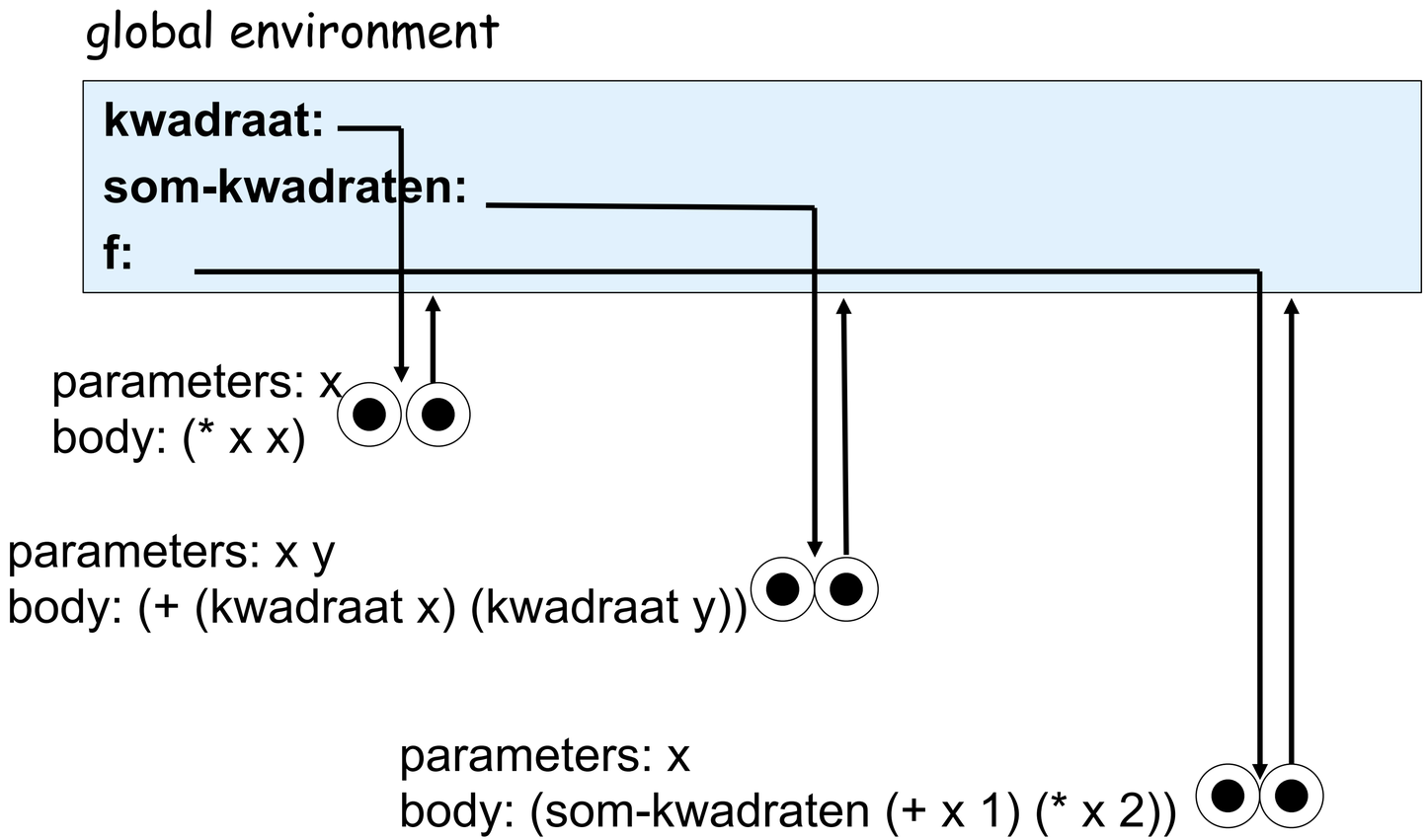
applicative order

(f 5)
(som-kwadraten (+ x 1) (x 2))*
(som-kwadraten (+ 5 1) (* 5 2))
(+ (kwadraat x) (kwadraat y))
(+ (kwadraat 6) (kwadraat 10))
(+ (x x) (* x x))*
(+ (* 6 6) (* 10 10))
(+ 36 100)
136

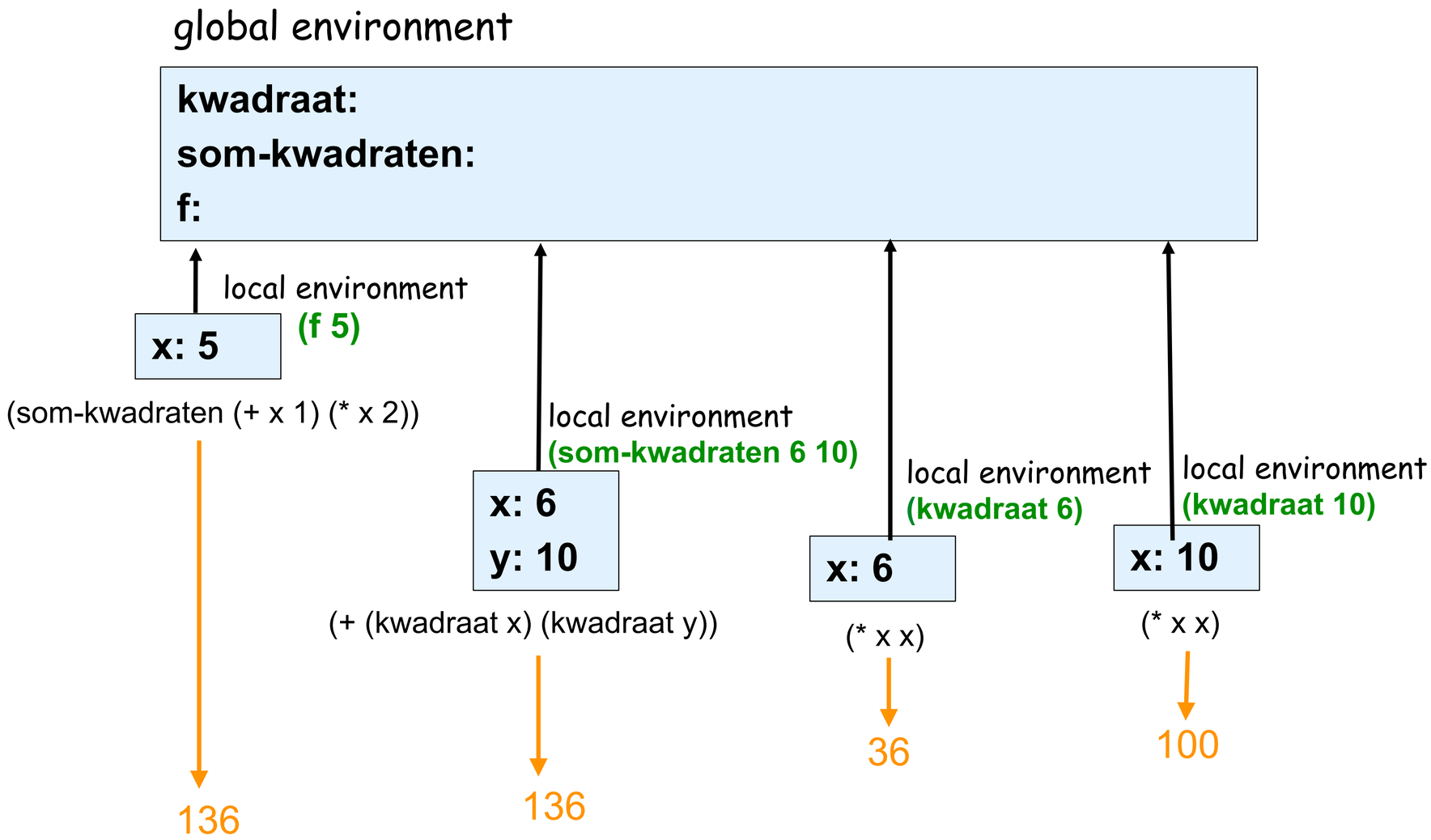
normal order

(f 5)
(som-kwadraten (+ x 1) (x 2))*
(som-kwadraten (+ 5 1) (* 5 2))
(+ (kwadraat x) (kwadraat y))
(+ (kwadraat (+ 5 1)) (kwadraat (* 5 2)))
(+ (x x) (* x x))*
(+ (* (+ 5 1) (+ 5 1)) (* (* 5 2) (* 5 2)))
(+ (* 6 6) (* 10 10))
(+ 36 100)
136

Het omgevingsmodel voor procedure evaluatie (1)



Het omgevingsmodel voor procedure evaluatie (2)



Procedures als zwarte dozen: Procedurele abstractie

```
> (define (dubbel x)
      (* x 2))
> (dubbel 5)
10

> (define (dubbel x)
      (+ x x))
> (dubbel 5)
10

> (define (dubbel n)
      (+ n n))
> (dubbel 5)
10
```

hoe dubbel is gedefinieerd is van geen belang voor de gebruiker (klant)

de naam van de formele parameter is totaal onbelangrijk

Les 2: Procedures en blokstructuren

SESSIE 1