

---

# Racket et POO

- 1 `#lang racket`
  - 2 (`define` `maclasse%` ; par convention, un nom de classe se termine par %
  - 3 (`class` `object%` ; hérite de la classe racine "object"
  - 4 ; constructeur
  - 5 ; x et y sont des arguments du constructeur
  - 6 (`init` `init_x` `init_y`)
-

---

# Racket et POO

7

8 ; attribut privé de la classe nommé x

9 ; sa valeur est initialisée avec l'argument init\_x du constructeur

10 (define x init\_x)

11 ; attribut privé de la classe nommé y

12 ; sa valeur est initialisée avec l'argument init\_y du constructeur

13 (define y init\_y)

---

---

# Racket et POO

14

15 ; appel du constructeur de la classe parente

16 (super-new)

17

18 ; méthode publique : renvoie la valeur de l'attribut x

19 (define/public (get\_x) x)

20 ; méthode publique : renvoie la valeur de l'attribut y

21 (define/public (get\_y) y)

---

# Racket et POO

22 ; méthode privée : additionne l'argument v à l'attribut x

23(define/private (ajout\_x v)

24. (set! x (+ x v)))

25 ; méthode privée : additionne l'argument v à l'attribut x

26 (define/private (ajout\_x v)

27 (set! x (+ x v)))

28 ;méthode privée : additionne l'argument v à l'attribut y

29 define/private (ajout\_y v)

30 (set! y (+ y v)))

---

# Racket et POO

31 ; méthode publique : addition avec un objet de même classe

32 (define/public (ajout o)

33. (ajout\_x (send o get\_x))

34. (ajout\_y (send o get\_y)))

---

---

# Racket et POO

35 ; méthode publique : affiche les attributs de l'objet

36 (define/public (affiche)

37. (display

38. (string-append "x = " (number->string x) " y = " (number->string y))))))

---

# Racket et POO

35 ; classefille hérite de maclasse

36 (define classefille%

37 (class maclasse%

38 (super-new)

39 ; surcharge de la méthode affiche

40 (define/override (affiche)(display

41 (string-append "x:" (number->string (send this get\_x))

42 "\n"

43 "y:" (number->string (send this get\_y))))))

---

# Racket et POO

44 ; création d'un objet de classe maclasse

45 (define o\_maclasse (new maclasse% [init\_x 10] [init\_y 20]))

46 ; création d'un objet de classe classefille

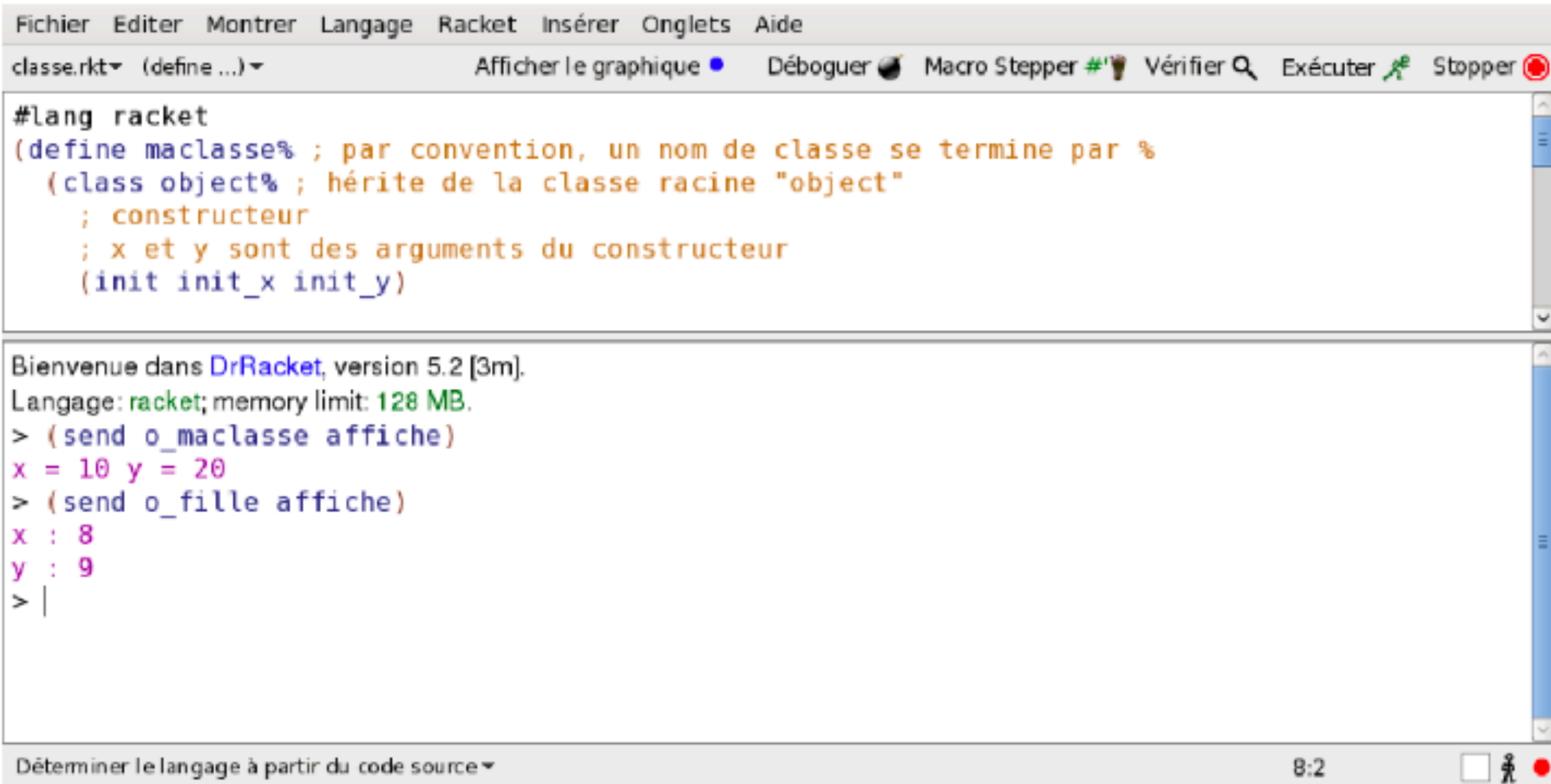
47(define o\_fille (new classefille% [init\_x 8] [init\_y 9]))

---



# Racket et POO

Exemple d'utilisation du code dans une session DrRacket



The screenshot shows the DrRacket IDE interface. The top menu bar includes 'Fichier', 'Editer', 'Montrer', 'Langage', 'Racket', 'Insérer', 'Onglets', and 'Aide'. Below the menu is a toolbar with buttons for 'Afficher le graphique', 'Déboguer', 'Macro Stepper', 'Vérifier', 'Exécuter', and 'Stopper'. The main editor window displays a Racket class definition for 'maclasse%'. The code defines a class that inherits from 'object%' and has a constructor 'init' that takes 'init\_x' and 'init\_y' as arguments. The bottom window shows the execution results, including a welcome message and the output of the 'send' commands.

```
Fichier Editer Montrer Langage Racket Insérer Onglets Aide
classe.rkt (define ...) Afficher le graphique Déboguer Macro Stepper Vérifier Exécuter Stopper

#lang racket
(define maclasse% ; par convention, un nom de classe se termine par %
  (class object% ; hérite de la classe racine "object"
    ; constructeur
    ; x et y sont des arguments du constructeur
    (init init_x init_y)

    ))

Bienvenue dans DrRacket, version 5.2 [3m].
Langage: racket; memory limit: 128 MB.
> (send o_maclasse affiche)
x = 10 y = 20
> (send o_fille affiche)
x : 8
y : 9
> |

Déterminer le langage à partir du code source 8:2
```