

# TD OCaml: objets avancés

David Delahaye

Faculté des Sciences  
[David.Delahaye@lirmm.fr](mailto:David.Delahaye@lirmm.fr)

Licence L3 2018-2019

# Exercice 1

## Types ouverts et fermés

Exercice à faire d'abord sur papier, puis en autocorrection avec OCaml.

```
class min (init : int) =  
object (self)  
  val x = init  
  method get = x  
  method min y = if self#get < y then self#get else y  
end;;  
  
class min_max (init : int) =  
object (self)  
  inherit min init  
  method max y = if self#min y = y then self#get else y  
end;;
```

# Exercise 1

## Types ouverts et fermés

```
class other =  
object  
  method get = 1  
  method min n = n - 1  
  method max n = n+1  
end;;  
  
class ['a] cell (init : 'a) =  
object  
  val mutable cont = init  
  method get = cont  
  method set n = cont <- n  
end;;
```

# Exercice 1

## Types ouverts et fermés

```
let m = new min 1;;  
let mm = new min_max 2;;  
let o = new other;;  
let c = new cell 1;;  
  
let natural (o: min) = o#min(0) = 0;;  
let negative (o: #min) = o#min(0) = o#get;;  
let positive o = o#get > 0;;
```

- 1 Quels sont les types des fonctions `natural`, `negative` et `positive`?

# Exercice 1

## Types ouverts et fermés

- ② Parmi les appels suivants, lesquels sont possibles ? Expliquer sa réponse et si un appel est possible, donner le résultat.

```
natural m;;  
natural mm;;  
natural o;;  
natural (o:>min);;  
natural c;;  
negative m;;  
negative mm;;  
negative o;;  
negative c;;  
positive m;;  
positive mm;;  
positive o;;  
positive c;;
```

## Exercice 2

### Classes virtuelles, héritage multiple, types ouverts

On souhaite modéliser les polynômes à une variable en modélisant, au préalable, toutes les structures algébriques nécessaires.

- 1 Définir les classes des magmas additif  $(M, +)$  et multiplicatif  $(M, \cdot)$ , où  $M$  est un ensemble et  $+/\cdot$  des opérations binaires sur ces ensembles. L'ensemble devra être un paramètre de ces classes et les classes correspondantes commenceront comme suit :

```
class virtual [ 'a ] add_magma = ...
```

```
class virtual [ 'a ] mul_magma =
```

## Exercice 2

### Classes virtuelles, héritage multiple, types ouverts

- ② Définir les classes des monoïdes additif et multiplicatif. Les monoïdes sont des magmas qui possèdent en plus un élément neutre pour les opérations correspondantes ( $+$  ou  $\cdot$ ). Les classes correspondantes commenceront comme suit :

```
class virtual ['a] add_monoid = ...
```

```
class virtual ['a] mul_monoid = ...
```

## Exercice 3

### Classes virtuelles, héritage multiple, types ouverts

- ③ Définir la classe des groupes additifs. Les groupes sont des monoïdes qui possède en plus une opération d'inverse. La classe correspondante commencera comme suit :

```
class virtual ['a] add_group = ...
```

- ④ Définir la classe des anneaux. Un anneau  $(R, +, \cdot)$  possède deux opérations  $+$  et  $\cdot$  telles que  $(R, +)$  est un groupe et  $(R, \cdot)$  est un monoïde. La classe correspondante commencera comme suit :

```
class virtual ['a] ring = ...
```



## Exercice 3

### Classes virtuelles, héritage multiple, types ouverts

- 5 Définir la classe `int_ring` représentant l'anneau des entiers ( $\mathbb{Z}$ ).
- 6 Définir la classe des polynômes à une variable sur un anneau quelconque. On rappelle qu'un polynôme  $P$  est de la forme :

$$P = a_n X^n + a_{n-1} X^{n-1} + \dots + a_1 X + a_0$$

où les  $a_i$  sont des éléments de l'anneau et  $X$  la variable.

On souhaitera avoir une méthode `eval` pour évaluer le polynôme lorsque la variable  $X$  prend une valeur dans l'anneau. La classe correspondante commencera comme suit :

```
class virtual ['a, 'b] polynomial  
  (r : 'b) (p : ('a * int) list) = ...
```

où  $r$  est la structure d'anneau et  $p$  la liste des couples (coefficients, puissance) du polynôme.

## Exercice 3

### Classes virtuelles, héritage multiple, types ouverts

- ⑦ Définir la classe des polynômes sur les entiers. Vous devrez au préalable créer une structure d'anneau sur les entiers. La classe correspondante commencera comme suit :  

```
class int_polynomial (p : (int * int) list) = ...
```
- ⑧ Définir le polynôme  $X^2 - 5X + 6$ , puis l'évaluer avec la valeur  $-3$ .
- ⑨ Modifier le code précédent pour que l'on puisse afficher les polynômes sur n'importe quel anneau.