TD OCaml: les objets

David Delahaye

Faculté des Sciences
David.Delahaye@lirmm.fr

Licence L3 2019-2020

Exercice 1

Liaison tardive et utilisation de `self`

Soit les classes suivantes :

```
class min (xi : int) =
object (self)
  val mutable x = xi
  method get = x
  method set n = x <- n
  method min y = if self#get < y then self#get else y
end;;

class min_zero xi =
object
  inherit min xi
  method get = 0
end;;
```

Exercice 1

Liaison tardive et utilisation de `self`

Pour chacun des envois de message suivants, expliquer :

1. Quelle est la réponse d'OCaml ?
2. Quelle méthode get est exécutée : celle de `min` ou celle de `min_zero` ?
3. Quelle est la valeur liée à `self` lors de chaque appel à `min` ?

```
let o1 = new min 4;;
let o2 = new min_zero 4;;
o1#min 2;;
o1#min 7;;
o2#min 2;;
o2#min (-2);;
```

Exercice 2

Collections et classes paramétrées

On considère les classes `account`, `interest_account` et `secure_account` vues en cours.

1. Écrire la classe `bank` avec les méthodes suivantes :
   - `add` : ajoute un compte bancaire ;
   - `balance` : calcule la somme des soldes des comptes ;
   - `print` : affiche les informations de tous les comptes ;
   - `fees` : prélève 5% de frais à tous les comptes.
2. Écrire la même classe mais en héritant d'une classe `collection` polymorphe (à écrire) qui permet de stocker des éléments de n'importe quel type.
   Quelles méthodes de la classe `bank` peuvent être « remontées » dans la classe `collection` ?

Exercice 3

Égalité structurelle et hiérarchie de classes

On souhaite modéliser les expressions arithmétiques (sur les entiers).

- Écrire les classes `cte`, `inv`, `add`, `sub`, `mul` et `div`, correspondant respectivement aux constantes et aux opérations d'inverse, d'addition, de soustraction, de multiplication et de division (quotient) avec les méthodes suivantes :
  - `eval` : évalue l'expression arithmétique ;
  - `print` : affiche l'expression arithmétique.
- Comparer les types `cte`, `inv`, `add`, `sub`, `mul` et `div`. Est-il nécessaire de construire une super-classe (que l'on pourrait appeler `expr`) à ces classes pour modéliser les expressions arithmétiques ?

Exercice 4

Arbres binaires de recherche (ABR)

Pour plus d'informations sur les ABR, voir :

https://fr.wikipedia.org/wiki/Arbre_binaire_de_recherche

- Écrire une hiérarchie de classes représentant les ABR dont les éléments sont d'un type quelconque et avec les méthodes suivantes :
  - `insert` : insère un élément dans un ABR ;
  - `find` : recherche un élément dans un ABR ;
  - `remove` : supprime un élément d'un ABR.