

Programmation applicative – L2

Troisième série de TP : Base de données de notes d'étudiants

G.Artignan {artignan@lirmm.fr} M.Lafourcade {lafourcade@lirmm.fr}
S.Daudé {sylvain.daude@univ-montp2.fr} A.Chateau {chateau@lirmm.fr}
B.Paiva Lima da Silva {bplsilva@gmail.com} C.Dony {dony@lirmm.fr}

1 Préambule

Cet énoncé concerne seulement 2 séances de TP. N'oubliez pas de sauvegarder précieusement votre travail au fur et à mesure des séances, pour pouvoir l'utiliser lors de l'examen final.

On s'intéresse à la modélisation d'une base de données pour stocker les notes que des étudiants ont eu à un examen. Nous allons nous intéresser aux processus de saisie des notes, récupération des notes, mais également accès divers à la base de données (moyenne, meilleures notes etc.). Un étudiant est identifié par un symbole unique, son nom, et obtient une et une seule note à l'examen, un entier compris entre 0 et 20.

2 Listes associatives

Nous allons utiliser la structure de donnée de liste associative ou a-liste pour modéliser cette base de données. Une a-liste est une liste de paires (clé . valeur). Ainsi, une entrée dans la base de données peut être considérée comme un couple [nom, note]. Voici l'interface de cette structure de données :

<code>empty-al</code>	La constante qui représente une a-liste vide.
<code>(empty-al? al)</code>	Le prédicat qui teste si al est une a-liste vide.
<code>(add-assoc k v al)</code>	Le constructeur qui ajoute l'association (k v) à la a-liste al si il n'existe pas d'association existante (sinon, remplace l'association existante). Ce constructeur retourne la nouvelle a-liste.
<code>(get-assoc k al)</code>	Renvoie l'association dont la clé est égale à k dans al (ou la liste vide si il n'existe pas d'association pour k).
<code>(get-keys al)</code>	Renvoie la liste des clés de al.
<code>(remove-assoc k al)</code>	Renvoie une a-liste contenant toutes les associations de al moins celle dont la clé est égale à k dans al.

Exercice 1 (long) *Implémentez chacune de ces fonctions, en testant chacune d'entre elles.*

Exercice 2 *Pour chacune de ces fonctions qui utilise une manipulation récursive de liste, donnez une version récursive terminale : par exemple `get-keys` aura une version `get-keys-term` avec la même signature, mais en récursivité terminale.*

3 Base de données

Dans la suite, on implémente une base de données de notes sous la forme d'une liste associative. Il faut donc en priorité se servir de l'interface déterminée ci-dessus.

Exercice 3 Ecrivez la fonction (saisir nom note bd) qui ajoute une entrée [nom note] à la base de données bd et renvoie la nouvelle base, en vérifiant que nom est bien de type symbole et note de type entier compris entre 0 et 20. Exemple de séquence de saisies de notes :

```
> (define b (saisir 'Domelevo 12 '()))
> (define b (saisir 'Legrand 11 b))
> (define b (saisir 'Dony 8 b))
> (define b (saisir 'Chateau 8 b))
> (define b (saisir 'Boudet 14 b))
> b
((Domelevo . 12) (Legrand . 11) (Dony . 8) (Chateau . 8) (Boudet . 14))
```

Exercice 4 Ecrivez la fonction (enlever nom bd) qui enlève une entrée [nom note] à la base de données bd et renvoie la nouvelle base. Exemple :

```
> b
((Domelevo . 12) (Legrand . 11) (Dony . 8) (Chateau . 8) (Boudet . 14) (Toto . 20))
> (define b (enlever 'Toto b))
> b
((Domelevo . 12) (Legrand . 11) (Dony . 8) (Chateau . 8) (Boudet . 14))
```

Exercice 5 Ecrivez la fonction (note nom bd) qui renvoie la note d'un étudiant nom dans la base bd. Exemple :

```
> (note 'Domelevo b)
12
```

Exercice 6 Ecrivez la fonction (etudiants bd note) qui renvoie la liste des étudiants ayant obtenu note à l'examen. Exemple :

```
> (etudiants b 8)
(Dony Chateau)
```

Exercice 7 Ecrivez la fonction (moyenne bd) qui calcule la moyenne des notes de la base de données bd. Exemple :

```
> (moyenne b)
10 3/5
```

Exercice 8 Ecrivez la fonction (ecart-type bd) qui calcule l'écart-type des notes de la base de données bd (Rappel, l'écart-type d'un échantillon x_1, x_2, \dots, x_n est $\sqrt{\frac{\sum_{i=1}^n (x_i - \hat{m})^2}{n}}$, où \hat{m} est la moyenne de l'échantillon). Exemple :

```
> (ecart-type b)
2.3323807579381204
```

Exercice 9 Ecrivez la fonction (combien bd) qui construit et renvoie une a-liste de paires [note nombre] qui précise pour chaque note présente dans bd combien d'étudiants ont eu cette note. Exemple :

```
> (combien b)
((12 . 1) (11 . 1) (8 . 2) (14 . 1))
```