

Lambda calcul et programmation fonctionnelle

TP 3

Nous allons écrire des fonctions permettant d'extraire des informations concernant des joueurs jouant à un jeu. Un joueur est un triplet de type (String, Int, Int), composé du pseudo du joueur, de son score et de son nombre de vies. Par exemple, la liste suivante représente l'état de tous les joueurs à un moment donné :

```
[("Toto", 17532, 3), ("Titi", 6541, 2), ("Tutu", 14731, 3),  
("Tata", 0, 3), ("Tete", 7931, 0), ("Tyty", 157, 1)]
```

Exercice 1

Écrivez le programme principal suivant :

```
main :: IO ()  
main =  
    let l = [("Toto", 17532, 3), ("Titi", 6541, 2), ("Tutu", 14731, 3),  
            ("Tata", 0, 3), ("Tete", 7931, 0), ("Tyty", 157, 1)] in  
        print l
```

Ce programme se contente d'afficher la liste (pas très intéressant).

Exercice 2

Écrivez une fonction `extractNom` qui prend un triplet en paramètre (représentant un joueur) et renvoie le premier élément (le nom). Utilisez le pattern matching pour cela.

Créez une fonction `queLesNoms` qui prend en paramètres une liste de joueurs et renvoie la liste des noms des joueurs. Par exemple, pour la liste ci-dessus, on renverra `["Toto", "Titi", "Tutu", "Tata", "Tete", "Tyty"]`.

Vous devrez utiliser la fonction `map`.

Modifiez votre programme principal et vérifiez que votre fonction fonctionne comme attendu.

Exercice 3

Écrivez une fonction `estVivant` qui prend un triplet en paramètre (représentant un joueur) et renvoie vrai uniquement si le troisième élément (le nombre de vies) est supérieur à zéro.

Créez une fonction `queLesNomsDesVivants` qui prend en paramètres une liste de joueurs et renvoie la liste des noms des joueurs qui ont au moins une vie restante. Par exemple, pour la liste ci-dessus, on renverra `["Toto", "Titi", "Tutu", "Tata", "Tyty"]`.

Vous devrez utiliser les fonctions `map` et `filter`. Vous aurez sans besoin de créer plusieurs fonctions.

Modifiez votre programme principal et vérifiez que votre fonction fonctionne comme attendu.

Exercice 4

Créez une fonction `meilleursJoueursVivants` qui prend en paramètres une liste de joueurs et renvoie le nom et le score du joueur ayant le score le plus élevé, parmi ceux qui ont au moins une vie. S'il y a plusieurs joueurs ex aequo, on les renverra tous.

Vous aurez sans doute besoin de créer plusieurs fonctions.

Modifiez votre programme principal et vérifiez que votre fonction fonctionne comme attendu.