

Feuille de travaux pratiques

Types

Exercice 1 (Jeu des types)

Ce jeu se joue en trois étapes. Lisez l'ensemble des règles avant de commencer.

Etape 1 Prenez une feuille de papier. Ecrivez dessus un type OCaml. Pliez la feuille en 4.

Etape 2 Passez la feuille à votre voisin.e (convenez d'un sens pour tout le groupe), et récupérez la feuille de votre autre voisin.e sans la déplier.

Etape 3 Quand toutes les feuilles ont été échangées, le jeu démarre. Dépliez la feuille et écrivez une expression en OCaml dont le type est celui de la feuille. La première ou le premier qui trouve a gagné !

A la fin du jeu, en vous appuyant sur les types qui ont été proposés dans le groupe, réfléchissez à ces questions : qu'est-ce qui caractérise un type difficile d'un type facile pour ce jeu ? Existe-t-il des types qui ne sont pas réalisables dans OCaml ?

Exercice 2 (Cartes)

Cet exercice a pour but de pratiquer les types en jouant aux cartes. On considèrera dans la suite un jeu de 52 cartes, qui ont pour valeur 2, 3, ... 9, 10, *Valet*, *Dame*, *Roi*, *As* dans les quatre couleurs *Pique*, *Coeur*, *Carreau*, *Trefle*.

1. Ecrire un type `face` qui représente la valeur d'une carte (de 2 à *As*).
2. Ecrire un type `couleur` qui représente la couleur d'une carte.
3. Ecrire un type `carte` qui représente une carte du jeu.
4. Définir les cartes `roipique` et `ascoeur`.

On a souvent besoin de comparer des cartes, par exemple pour savoir qui remporte un pli. Ici, on suppose que la carte ayant la valeur la plus forte est gagnante (comme à la bataille par exemple).

5. Ecrire une fonction `compare` qui compare deux cartes : on réfléchira soigneusement à la fois à la méthode (il s'agit de comparer les valeurs, qui sont données sous la forme de noms), au typage, et à la définition éventuelle des fonctions auxiliaires utiles.
6. Ecrire une fonction `tire_random` qui tire une carte aléatoirement dans le paquet de 52 cartes. Ecrire une fonction `test` qui teste cette fonction sur au moins 100 tirages aléatoires, et tester.
7. Ecrire une fonction `score_carte` qui attribue un score à une carte de la façon suivante : un as gagne 11 points, un 10 gagne 10 points, un roi 4 points, une dame 3 points, un valet 2 points et les autres cartes ne gagnent rien.

8. Ecrire une fonction `bataille` qui implémente un (faux) jeu de bataille fonctionnant de la façon suivante : on a deux joueurs. A chaque tour, chacun "retourne" (tire aléatoirement) une carte, on compare les valeurs des deux cartes, et le joueur qui a retourné la carte gagnante marque le nombre de points correspondant aux deux cartes remportées (la sienne et celle de son adversaire).
9. Pourquoi ce jeu n'est-il pas équivalent à la vraie bataille ? Quel ingrédient nous manque-t-il (dans le langage) pour pouvoir faire un jeu de bataille plus réaliste ?