

# Introduction à la programmation fonctionnelle

## Cours 7

kn@lri.fr

<http://www.lri.fr/~kn>

# Au menu

---

On va tenter de modéliser un programme qui simule le jeu « le compte est bon »

- ◆ Analyse descendante
- ◆ Réflexion sur les structures de données et algorithmes

Le code des fonctions sera ensuite à faire en TP.

On utilisera évidemment une approche *fonctionnelle* en raisonnant en terme de fonctions et de leur type.

# Plan

---

- 1 IPF (1) : expressions de bases, if/then/else, fonctions ✓
- 2 IPF (2) : fonctions récursives, inférence de types ✓
- 3 IPF (3) : Types structurés, filtrage, polymorphisme, ordre supérieur ✓
- 4 IPF (4) : Exceptions, Listes (1) ✓
- 5 IPF (5) : Fonctions anonymes, Itérateurs, Application partielle ✓
- 6 IPF (6) : Algorithmes avancés sur les listes ✓
- 7 IPF (6) : Algorithmes avancés sur les listes

## 7.1 Le compte est bon

# Description du jeu

Dans ce jeu, un oracle choisit aléatoirement un entier compris entre 100 et 999 (par exemple 425)

On dispose ensuite de 24 cartes :

1	1	2	2	3	3	4	4	5	5	6	6	7	7
8	8	9	9	10	10	25	50	75	100				

Les cartes sont retournées et 6 sont choisies au hasard, par exemple :

2	4	4	7	10	50	75
---	---	---	---	----	----	----

Deux joueurs ont alors un temps limité pour trouver l'entier cible en utilisant les cartes au plus une fois et en les combinant avec les opérations +, -, ×, ÷ :

$$((2 \times 50) \times 4) + (75 / (7-4)) = 425$$

## Description du jeu (2)

---

Règle pour les opérations :

- ◆ On peut faire des additions et des multiplications librement
- ◆ On ne peut soustraire deux nombres que si le résultat est strictement positif
- ◆ On ne peut diviser deux nombres que si la division est exacte (le reste est nul)

# Description du programme

---

On souhaite écrire un programme OCaml qui simule ce jeu :

- ◆ Le programme choisit un nombre aléatoire
- ◆ Le programme choisit 6 cartes aléatoirement
- ◆ Le programme recherche une solution et l'affiche

Nombre cible: 278

Cartes: 50 5 8 75 7 5

Solution:

$$\Rightarrow 50 / 5 = 10$$

$$\Rightarrow 10 + 8 = 18$$

$$\Rightarrow 75 - 18 = 57$$

$$\Rightarrow 57 * 5 = 285$$

$$\Rightarrow 285 - 7 = 278$$

Attention, il n'existe pas toujours une solution.

# Analyse descendante

---

Dans l'approche par analyse descendante, on écrit le programme en partant de la fonction principale.

On fait appel à des fonctions plus élémentaires, que l'on ajoute alors au programme.

On répète ensuite l'opération pour chaque fonction introduite jusqu'à arriver à des fonctions élémentaires.

Pour ce cours, on va écrire les fonctions jusqu'à arriver au niveau élémentaire. Il faudra ensuite compléter ces fonctions en TP dans le fichier `1ceb.ml`.