

Langage Python

TD 3 : Paradigmes de programmation

Hubert Godfroy

12 novembre 2015

1 Programmation orientée objet

Question 1 :

On souhaite modéliser une population d'individus. Une `Personne` dispose d'un nom, d'un âge, d'un poids, d'une taille,

Proposer une classe `Personne`.

Question 2 :

Écrire un constructeur créant une personne en lui associant un nom.

Question 3 :

Écrire une méthode donnant le nom d'une `Personne`.

Question 4 :

Créer deux classes `Homme` et `Femme` héritant de la classe `Personne`.

Un `Homme` et une `Femme` ont chacun un père et une mère.

Question 5 :

Écrire une fonction associant à un père et une mère à un individu. La fonction devra refuser de faire d'éventuelles changements si le père ou la mère sont déjà définis.

Question 6 :

Écrire une fonction donnant les parents d'un individu.

Question 7 :

Écrire une fonction donnant les grands-parents d'un individu.

Question 8 :

Écrire une fonction donnant l'ancêtre commun le plus proche de deux individus.

Question 9 :

Un `Homme` peut avoir une épouse et une `Femme` un mari : modifier les classes respectives afin de prendre en compte cette information.

Question 10 :

Écrire une fonction `mariage` associant à un `Homme` une `Femme` et à une `Femme` un `Homme`.

Question 11 :

Écrire une fonction cherchant s'il existe un lien de parenté (naturel ou par alliance) entre deux individus.

2 Programmation fonctionnelle

2.1 Compréhension

Les questions de cette section doivent être résolues en utilisant les compréhensions.

Question 12 :

Construire la liste des nombres pairs de 1 à 100.

Question 13 :

Construire la liste des nombres premiers de 1 à 100.

2.2 Fonctions comme valeurs de première classe

Question 14 :

Construire une fonction calculant la valeur en 0 d'une fonction d'entiers.

Question 15 :

Construire une fonction prenant une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ et un tuple d'entier (n_1, \dots, n_k) et créant le tuple $(f(n_1), \dots, f(n_k))$

Question 16 :

Construire une fonction prenant un prédicat sur les entiers et un ensemble d'entiers et construisant l'ensemble des entiers de l'ensemble vérifiant ce prédicat.

Un prédicat sur les entiers est une fonction des entiers vers les booléens.

Question 17 :

On se propose de trouver un nouvel encodage des entiers. Au lieu d'utiliser les nombres entiers habituels, on va les définir en utilisant uniquement des fonctions. On utilise pour cela la représentation de CHURCH.

Si n est un entier, sa représentation fonctionnelle est la fonction φ_n telle que, si $f : \mathbb{N} \rightarrow \mathbb{N}$ est une fonction,

$$\varphi_n f = \underbrace{f \circ f \circ \dots \circ f}_{n \text{ fois}}$$

- (a) Construire les entiers de CHURCH correspondant à 1, 2, 3, 0.
- (b) Construire une fonction passant des entiers normaux aux entiers de CHURCH.
- (c) Construire la fonction réciproque.
- (d) Construire la fonction sommant deux entiers de CHURCH.
- (e) Construire la fonction multipliant deux entiers de CHURCH.

Question 18 :

On souhaite maintenant réécrire les booléens et l'opérateur `if` en utilisant uniquement un style fonctionnel.

On pose `true` (resp. `false`) la fonction prenant en argument `x` et `y` et retournant `x` (resp. `y`).

- (a) Construire les fonctions `true` et `false`.
- (b) Construire une fonction `if` prenant en paramètre une fonction et deux objets, et retournant le premier objet si la fonction est `true` et le second si la fonction est `false` :

$$\begin{aligned} \text{if}(\text{true}, a, b) &\mapsto a \\ \text{if}(\text{false}, a, b) &\mapsto b \end{aligned}$$