

Bases de la programmation impérative

Ensimag 1A

1. Opérations sur les listes

Dans cet exercice, on se propose de travailler sur des opérations de haut niveau sur des listes. Nous utiliserons les mêmes types que lors du TD précédent.

Pour l'ensemble de l'exercice, à part le constructeur, aucune méthode ne doit créer de cellules.

- 1.1. Modifiez le constructeur `__init__` de la classe `Liste` pour construire une liste à partir d'un itérable. Par exemple `Liste(range(1, 10))` doit construire une liste contenant tous les entiers entre 1 et 10.
- 1.2. Implémentez une méthode `transformation(self, fonction)` qui remplace pour chaque cellule la valeur par `fonction(valeur)`.
- 1.3. Étant donnée une fonction `fonction` à valeurs booléennes, on souhaite itérer sur toutes les cellules de la liste dont la valeur vérifie `fonction(valeur)`. Implémentez une méthode `iterateur_filtre(self, fonction)` renvoyant un tel itérateur.
- 1.4. Utilisez votre itérateur pour implémenter une méthode `filtre(self, fonction)` qui élimine toutes les cellules dont la valeur *ne* vérifie *pas* `fonction(valeur)`.
- 1.5. Implémentez une méthode `concatenation(self, liste_fin)` rajoutant les cellules de la liste `liste_fin` à la fin de la liste `self`. La liste `liste_fin` doit devenir vide.
- 1.6. Transformez une liste en mettant au début tous ses éléments pairs puis tous ses éléments impairs. L'ordre des éléments pairs (resp. impairs) entre eux est conservé.