

Des questions pour se préparer au CC

Dernière mise à jour : le 2 février 2019

Algorithmique

Exercice 1. Montrez que l'algorithme de parcours en profondeur récursif ne se comporte pas exactement comme l'algorithme de parcours en profondeur qui utilise une pile.

Pour ce faire on choisira un graphe (non orienté) de petite taille, et

1. On détaillera l'exécution de l'algorithme de parcours en profondeur sur ce graphe, étape après étape. On décrira clairement l'arbre recouvrant construit par l'algorithme.
2. On détaillera l'exécution sur ce graphe de l'algorithme de parcours en profondeur récursif, étape après étape, et on décrira clairement l'arbre recouvrant construit par cet autre algorithme.
3. On observera que les deux arbres recouvrant ne sont pas les mêmes.

POO

Exercice 2. Nous avons approché, en cours, la programmation orienté objet. On souhaite tester votre compréhension des principes fondamentaux de cette approche à la programmation.

1. Expliquez ce que c'est l'héritage.
2. Apportez des exemple (depuis ce qu'on a vu en cours où en TP, ou d'autres exemples) d'héritage.
3. Comment peut on profiter de l'héritage lors de la programmation ?
4. Quel est le lien entre héritage et polymorphisme ?

Structures de données

Exercice 3. Nous avons vu, en cours, différentes implémentations de la structure de données abstraite *Set*.

1. Rappelez quelles sont ces implémentations.
2. Comparez ces implémentations en précisant leur particularités.

Un peu de programmation

Exercice 4. En utilisant le langage **python**, décrivez l'interface d'une classe dont on se servira pour représenter des polynômes. Ajoutez à cette classe autant des méthodes qu'il vous intéresse d'ajouter, n'oubliant pas de donner une description de ce que ces méthodes sont sensé réaliser.

Exercice 5. Esquissez l'implémentation d'une classe pour représenter les polynômes. On pourra, par exemple, représenter un polynôme par la liste des ses coefficients.

Voici l'interface de la classe qu'on vous demande d'implémenter :

```
class PolynomeAbstract():
    """Interface de la classe Polynome"""

    def __init__(self):
        self.coeficients = []

    def evaluer(self, value):
        """Evaluer un polynome p(x) avec x = value"""

    def degree(self):
        """Retourne le degre d'un polynome"""

    def square(self):
        """Retourne p(x)^2"""
```