

Intervalles

Travaux Pratiques 01 du module 06 – La POO

Avant de démarrer ce TP, il convient d'avoir suivi les modules 1 à 6 de ce cours.

Durée estimée

1h à 1h30

Énoncé

L'objectif de cet exercice est de créer des intervalles, délimités par une borne minimale et une borne maximale, afin de pouvoir pratiquer dessus un certain nombre d'opérations.

Un fichier de test est disponible pour vérifier votre proposition.

1. Créer une classe `Intervalle` possédant un constructeur avec deux paramètres : une borne inférieure et une borne supérieure. Vous devez vérifier que ces bornes sont dans le bon ordre, et sinon les inverser.
2. Créer la méthode d'affichage d'un intervalle. `print(i)` affiche `[4 ; 8]`, où 4 est la borne inférieure, 8 la borne supérieure.
3. Faites en sorte que les deux attributs ne soient pas visibles depuis l'extérieur de la classe.
4. Créer les setters permettant de modifier les bornes de l'intervalle : celle-ci doit vérifier que la borne inférieure reste toujours inférieure à la borne supérieure. Si ce n'est pas le cas, elle doit afficher un message d'erreur, et ne pas procéder à la modification. De même si la borne max est inférieure à la borne min.
5. Ecrire les getters correspondants
6. Redéfinir la méthode `__add__(self, other)` qui retourne un nouvel `Intervalle` addition des deux intervalles. Exemple : `[2 ; 5] + [3 ; 4] = [5 ; 9]`

7. Ecrire une méthode `intersection(self, other)` qui retourne l'intersection des deux intervalles et « None » si leur intersection est vide.
Exemple : $[2 ; 5] \cap [3 ; 6] = [3 ; 5]$
8. Lorsque l'intersection de deux intervalles n'est pas nulle, on peut procéder à leur union. Ecrire une méthode `union(self, other)`, qui renvoie None lorsque l'union est impossible, et le résultat autrement.
Exemple : $[2 ; 5] \cup [3 ; 6] = [2 ; 6]$

Solution

Des propositions de solution pour ce TP sont placées dans les éléments en téléchargement liés à ce module.