

But du TD

Implémenter une classe simple (sans pointeurs et sans héritage).

Ce que vous devez avoir compris à la fin du TD:

- **Obligatoire:** Le principe d'encapsulation des données membres (private-public-"getters"-
"setters").
- **Obligatoire:** Savoir créer le Header d'une classe en utilisant les bonnes déclarations de fonctions.
- **Obligatoire:** Les différents constructeurs (vide, avec paramètres, par récopie).
- **Semi Avancé:** La surcharge d'opérateurs comme l'addition.
- **Avancé:** La surcharge de l'opérateur "==".

La classe interval

Un interval est simplement la combinaison d'une borne inférieure et d'une borne supérieure (par exemple [-3, 1] est un interval). Nous aimerions créer des objets "interval" afin de réaliser des opérations comme des Unions, des Intersections...

Tache 1

Réalisez le Header de la classe interval ayant comme données membres:

- `Borne_sup` (choisissez le type)
- `Borne_inf` (choisissez le type)

Et comme fonctions membres:

- Des constructeurs:
 - Constructeur vide qui initialise à [0,0]
 - Constructeur avec paramètre qui initialise à [a,b]
 - Constructeur par recopie
- Une fonction d'affichage qui affiche : "l'interval vaut [a,b]" (avec les bonnes valeurs bien sûr)
- Une fonction largeur qui calcule la largeur d'un interval

La Portée Public

Dans un premier temps, mettre la portée des données membres à `Public`.

Que se passe t'il dans le `main` suivant:

```
Interval I(3,5);  
I.Affiche();  
I.borne_inf=7;  
I.Affiche();
```

Pourquoi cela pose problème? Comment corriger cela?

La Portée Private

Passer les données membres en private. Créer les fonctions nécessaires :

- les *"getters"* : `get_borne_inf`, `get_borne_sup`
- les *"setters"* : `set_borne_inf` , `set_borne_sup`

Réécrire le `main` précédent, qu'affiche le programme?

S'exercer seul

- Créer dans le `main`, un tableau d'interval, et modifier les valeurs de chacun de ses éléments
- (semi-advanced) Créer dans la classe `interval`, une fonction **Union**, lorsque l'on execute : `IC = IA.Union(IB);` (IA, IB et IC sont des `interval`), alors
 - `IC.borne_inf=min(IA.Borne_inf, IB.Borne_inf)` et `IC.borne_sup=min(IA.Borne_sup, IB.Borne_sup)`
- (advanced) Créer dans la classe `interval` une surcharge de l'opérateur `+`
- (advanced) Créer dans la classe `interval` une surcharge de l'opérateur `=`