



HLIN302 – Travaux Dirigés n° 8

Programmation impérative avancée
Alban MANCHERON et Pascal GIORGI

Rappel

Dans les précédents TD, nous avons défini plusieurs classes (Itv, Cellule, Population, PopulationVivante, TableauCellule, Option, TabOptions, JDV, ...). Systématiquement, nous avons dû définir des fonctions ou des méthodes pour afficher le contenu des objets ou bien les initialiser, accéder à certains attributs (notamment l'accès individuel aux éléments des attributs de type tableaux), comparer des objets entre eux, ... Il nous manquait le sucre syntaxique de la surcharge pour rendre toutes ces opérations encore plus intuitives (et agréables) à utiliser.

1 Retour sur la classe `Itv` (cf TD 2)

Fichier `itv.h`

```
1  class Itv {
2  private:
3      // Attributs
4      double bi, bs;
5
6
7  public:
8      // Constructeurs
9      Itv();
10     Itv(double bi, double bs);
11
12     // Accesseurs en lecture
13     double getBorneInf() const;
14     double getBorneSup() const;
15
16     // Accesseurs en écriture
17     void setBorneInf(double v);
18     void setBorneSup(double v);
19
20     // Autres Méthodes
21     void Afficher() const;
22     double Longueur() const;
23     bool Appartient(double v) const;
24
25     // Nouvelles Méthodes
26     bool estEgal(const Itv &I) const;
27     bool estInclusStrictement(const Itv &I) const;
28     bool estDisjoint(const Itv &I) const;
29     bool estAccole(const Itv &I) const;
30     bool estImbrique(const Itv &I) const;
31
32     void Translate(double v);
33 };
34
```

```
35 void Translate(Itv &I, double v);  
36 #endif
```

1. Surcharger les opérateurs d’entrée/sortie (<< et >>) permettant d’afficher / de saisir un intervalle.
2. Surcharger les opérateurs de comparaison == et !=.
3. Est-il possible de remplacer les méthodes `estInclusStrictement`, `estDisjoint`, `estAccole` et `estImbrique` respectivement par les opérateurs <, >, <= et >= ?
4. Est-il possible de remplacer la méthode `Translate` par l’opérateur += ?
5. Même question pour la fonction `Translate` ?
6. Quelles autres surcharges d’opérateur vous semblent intéressantes pour cette classe ?

2 Jeu de la vie en plus joli

1. Surcharger les opérateurs d’entrée/sortie pour la classe `Cellule`.
2. Surcharger l’opérateur de sortie de la classe `Population` (cf TD 4 et suivants).
3. Surcharger l’opérateur de sortie de la classe `Option` (cf TD 4).
4. Surcharger l’opérateur de sortie de la classe `TabOptions` (cf TD 4).
5. Surcharger l’opérateur += de la classe `TabOptions` (au lieu de `addOption`).
6. Surcharger les opérateurs d’entrée/sortie de la classe paramétrée `MyVector` (cf TD 7).
7. Surcharger les opérateurs de comparaison entre `Cellules` (cf TD 6).
8. Surcharger les opérateurs paramétrés d’accès aux éléments de la classe `MyVector` (cf TD 7).
9. Surcharger les opérateurs de pré- et post-incrémentation de la classe `Population` (au lieu de `next`).
10. Surcharger l’opérateur () de la classe `TabOptions` (au lieu de `getOptionIndex`).
11. Surcharger l’opérateur de transtypage vers `int` de la classe `Option` (qui transforme l’option en son identifiant).