

Pseudo-langage

Les variables sont indiquées avec leur type :

booléen *b*, entier *n*, réel *x*, caractère *c*, chaîne *s*, etc.

On est souple du moment qu'il n'y a pas d'ambiguïté

Le signe de l'affectation est ←.

Si *A* est un tableau, *A*[*i*] est le *i*ème élément du tableau.

Si *P* est une structure (ou un objet), *P*.*x* est le champ (ou attribut) *x* de *P*.

Les instructions simples sont séquencés par « ; » (si besoin)

Les blocs d'instructions sont entourés par { ... }, **début** ... **fin**, ou simplement indiqués par la mise en forme (même niveau d'indentation).

La conditionnelle est exprimée par :

```
si condition
    instruction1
sinon
    instruction2
```

L'itération est exprimée par :

```
tant que condition
    instruction

faire
    instruction
tant que condition

répéter
    instruction
jusqu'à condition

pour i de min à max
    instruction
```

Les fonctions sont déclarées avec la syntaxe :

```
maFonction(↓ int i, ↑ int j, ↑ int k);
```

↓ = en entrée, ↑ = en sortie, ↑ = en entrée/sortie ; par défaut, on considérera que le paramètre est passé en entrée.

Les classes sont déclarées avec la syntaxe :

```
classe MaClasse étend SuperClasse réalise MonInterface
```

Les méthodes d'une classe sont déclarées en utilisant la même syntaxe que les fonctions. Si x est un objet, l'invocation de la méthode $m()$ sur x est dénotée avec la syntaxe

$x.m()$

Le constructeur est une méthode spéciale d'une classe, dont le nom est le même que celui de la classe. On pourra aussi l'appeler, plus explicitement, **constructeur**()).

Si « MaClasse » est le nom d'une classe, on peut dénoter la création d'un nouveau objet instance de la classe avec la syntaxe :

nouveau MaClasse()

ou, plus simplement,

MaClasse()