

Cartographie de la grammaire C++

L'approche didactique proposée repose sur une décomposition en trois sections complémentaires du langage C++.

Fondamentaux algorithmiques de la programmation

Sur un premier niveau, nous trouvons tout ce qui concerne l'écriture algorithmique et ses fondamentaux, à savoir : variables, opérations, structures de contrôle (if, switch, boucles, fonctions), ensembles de variables (structures et tableaux) et pointeurs. En bref, il s'agit du langage C intégré au langage C++.

Fonctionnalités diverses et correctifs

Sur le même niveau, le C++ apporte des élargissements fonctionnels et des correctifs afin de faciliter la pratique (la surcharge des fonctions par exemple) et d'introduire plus de rigueur (meilleur contrôle des conversions de types).

Ce sont des outils propres au C++, souvent communs avec le C#, qui simplifient l'écriture ou proposent des fonctionnalités, mais sans intervenir du point de vue de la modélisation sur la structuration globale d'un programme.

À ce premier niveau grammatical et syntaxique, le C++ n'intervient pas encore profondément sur la structuration globale d'un programme.

Outils de structuration et d'organisation des données et des traitements

Sur un second niveau se trouve l'essentiel du C++ : des outils de structuration qui vont permettre de donner forme à un programme. Il s'agit d'outils sur lesquels repose l'architecture objet des programmes ; inévitablement, ils nous font entrer dans une logique de conception et de modélisation. Nous trouvons :

Classe, objets, problématiques de relations entre objets, héritage, problématique du polymorphisme avec virtualité, classe abstraite et interface.

Sur ce même niveau se trouve l'ensemble des fonctionnalités associées à ces différents concepts, par exemple toutes les fonctions spéciales dans une classe (constructeurs, destructeur), mais aussi les surcharges de l'affectation (copie et déplacement) et des opérateurs en général, le pointeur this, etc.

Patrons de conception (design pattern) et modélisation UML

Une suite possible, qui n'est pas abordée dans ce livre, consisterait à étudier les patrons de conception et à acquérir quelques notions d'UML. Les patrons de conception répondent à des problèmes de conception de logiciels dans le cadre de la programmation par objet. Ce sont des solutions connues et éprouvées dont la conception provient de l'expérience des programmeurs. UML, pour "Unified Modeling Language", est un langage de modélisation graphique à base de pictogrammes, très utile au développement logiciel et à la conception orientée objet.