

Organisation du livre

Le livre se décompose en deux grandes parties. La première s'attache surtout à un premier niveau du langage. Elle regroupe ce qui concerne les fondamentaux algorithmiques hérités du langage C avec les améliorations et les correctifs propres au C++. La seconde partie développe les principes syntaxiques de la programmation objet et de ses fonctionnalités.

Dans la première partie, il s'agit de s'emparer des éléments primordiaux de la syntaxe algorithmique, indispensables pour entrer dans la dimension objet du C++ qui apparaît avec les structures. Toutes les fonctionnalités additionnelles du C++ sont utilisées et intégrées dans la réalisation de nos programmes au fur et à mesure de leur introduction. Très rapidement, dès le début de l'étude, nous dépassons les possibilités syntaxiques du langage C. Les points suivants sont abordés :

Le chapitre Premiers programmes récapitule brièvement la relation entre C et C++ et il expose les principes les plus basiques de la mise en œuvre de l'écriture d'un programme : la fonction main, les commentaires, les inclusions, l'affichage d'un texte. Deux premiers programmes sont réalisés : "hello world!" en C et "hello world" en C++.

Le chapitre Variables simples, constantes, affectation expose ce qui concerne les variables, les constantes et les affectations de valeurs.

Le chapitre Affichage et saisie console montre comment afficher et saisir des valeurs dans une fenêtre console.

Le chapitre Opérations porte sur les opérations. Nous y traitons la notion d'expression, les opérations arithmétiques, l'obtention de valeur aléatoires, les opérations bit à bit.

Le chapitre Conversions de types porte sur les conversions de types nécessaires dans certaines opérations entre variables de types différents.

Le chapitre Structures de contrôle s'intéresse aux structures de contrôle : sauts conditionnels (if-else), branchements (switch) et boucles (while, do-while, for).

Le chapitre Fonctions montre comment écrire ses propres fonctions en utilisant ou pas des paramètres et une valeur de retour. Les apports du C++, valeurs par défaut de paramètres, surcharge et fonctions génériques, sont également étudiés.

Le chapitre Portée et durée de vie des variables détaille la portée et la durée de vie des variables, il évoque notamment les possibles masquages.

Avec le chapitre Structures sont introduites les prémisses de l'objet, à savoir la structure synonyme de classe en C++.

Le chapitre Bibliothèques et espaces de noms récapitule ce qui concerne la création de bibliothèques et d'espaces de noms.

Le chapitre Unions examine le principe de l'union. On y trouve en particulier une étude d'union discriminée.

Le chapitre Tableaux statiques traite les tableaux statiques. Il propose également une introduction aux conteneurs dont l'utilisation est indispensable et précieuse en C++.

Le chapitre Chaînes de caractères étudie les chaînes de caractères et l'utilisation de la classe string.

Le chapitre Pointeurs expose ce qui concerne les pointeurs et les références. Nous y trouvons les opérateurs associés, les fonctions d'allocation, le pointeur générique void*, les valeurs NULL et nullptr, la notion de constante associée aux pointeurs, le type référence, les opérateurs new et delete.

Dans le chapitre Principales utilisations des pointeurs nous abordons les pointeurs en paramètre de fonctions (passage par référence) l'allocation dynamique, l'allocation dynamique de tableaux et une introduction aux listes chaînées.

Le chapitre Gestion des erreurs présente les techniques et les outils dédiés à la gestion des erreurs et des exceptions.

Le chapitre Deux automates cellulaires qui clôt la première partie du livre propose deux études de cas, deux automates cellulaires : la simulation d'un feu de forêt et le fameux automate cellulaire de John Conway.

La seconde partie du livre approfondit la question de l'objet et se focalise sur les moyens de structuration objet apportés par le C++. Dans cette partie, chaque chapitre est suivi d'une section de mise en pratique qui propose des exercices spécifiques. Peut-être un trait d'originalité de ce livre car des exercices simples qui portent sur la mise en pratique de la dimension objet restent assez rares la plupart du temps.

Le premier chapitre de la seconde partie, le chapitre Classes, objets, détaille les notions de classe et d'objet. En partant d'une étude de cas il amorce une réflexion sur la conception objet puis détaille l'ensemble des fonctionnalités qui sont associées. Nous trouvons les droits d'accès, le pointeur this, l'ensemble des fonctions spéciales (constructeurs, constructeurs de copie, de déplacement, surcharges de l'opérateur = pour la copie et le déplacement, destructeur), les mots clés default et delete (C++2011), la notion d'accesseurs, l'instruction friend.

Le chapitre Surcharge des opérateurs s'attache à explorer la surcharge des opérateurs. En particulier nous découvrons comment transformer un objet en une fonction grâce à la surcharge de l'opérateur de fonction : ().

Le chapitre Classes et membres static s'occupe des éléments static dans une classe. Notamment c'est l'occasion d'une étude de cas basée sur un patron de conception : le singleton.

Le chapitre Classes génériques expose principes et fonctionnalités d'une classe générique.

Le chapitre Classe d'objets changeants qui suit porte sur les classes d'objets changeants (unions illimitées C++ 2011).

Le chapitre Associations entre objets traite les associations entre objets : échanges de messages, agrégations et coopérations entre objets, compositions et dépendances entre objets.

Le chapitre Héritage examine la notion d'héritage avec les relations base-dérivées, les droits d'accès de l'héritage, l'héritage multiple, l'héritage multiple avec base virtuelle, la distinction entre un héritage et une association.

Le chapitre Polymorphisme et virtualité traite la question d'un polymorphisme de pointeur avec les fonctions virtuelles et leurs redéfinitions.

Le chapitre Classe abstraite et interface considère avec les fonctions virtuelles pures le principe des classes abstraites et se clôt sur la notion d'interface.

Dans le dernier chapitre Quelques comparaisons entre C et C++ il ne s'agit pas de discuter du bien fondé de la programmation objet par rapport à la programmation C classique mais plutôt de réaliser que de mêmes outils peuvent prendre un sens différent et s'utiliser autrement dans chacun des deux langages. Nous nous sommes penchés tout particulièrement sur l'usage des fonctions et des pointeurs, les emplois d'unions et de typedef sont également abordés.