

*mmmm*

**comprendre et utiliser**  
**C++**  
**pour programmer objets**

**G. CLAVEL I. TRILLAUD L. VEILLON**



MASSON

**MANUELS INFORMATIQUES MASSON**



# **comprendre et utiliser C++ pour programmer objets**

**Gilles CLAVEL**

*Directeur consultant de la société IMA-Informatique  
Professeur à l'Institut National Agronomique*

**Isabelle TRILLAUD**

*Responsable études et projets de la société IMA-Informatique*

**Luc VEILLON**

*Ingénieur de recherche  
Directeur du centre de calcul de l'Institut National Agronomique*

**MASSON**

Paris Milan Barcelone

1994

# Table des matières

---

<b>Avant-propos</b> .....	IX
<b>Avertissement au lecteur</b> .....	XI
<b>1 Des techniques de base aux mécanismes objets</b> .....	1
1.1 - Simulation d'un distributeur.....	1
1.2 - Définir des variables, affecter des valeurs.....	2
1.3 - Schémas itératifs et conditionnels.....	3
1.3.1 - <i>L'énoncé while</i> .....	3
1.3.2 - <i>Structure de bloc, énoncés for et do ... while</i> .....	4
1.3.3 - <i>Expression d'un choix</i> .....	6
1.4 - Utilisation de fonctions, structure d'un programme.....	7
1.4.1 - <i>La fonction main</i> .....	7
1.4.2 - <i>Définition et déclaration de fonctions</i> .....	8
1.4.3 - <i>Le corps de la fonction main</i> .....	9
1.4.4 - <i>Plusieurs fonctions pour un programme</i> .....	10
1.4.5 - <i>Utilisation de la valeur renvoyée par une fonction</i> .....	11
1.4.6 - <i>Programme composé d'un seul module</i> .....	12
1.4.7 - <i>Programme composé de plusieurs modules</i> .....	13
1.5 - Objets et messages.....	18
1.6 - Notion de classe.....	20
1.7 - Données membres et fonctions membres.....	22
1.8 - Accès public ou privé.....	24
1.9 - Utilisation d'un constructeur.....	26
1.10 - Allocation dynamique.....	27
1.11 - Une simulation élémentaire.....	29
<b>2 Techniques de base en C++</b> .....	33
2.1 - Une classe pour enregistrer des entiers.....	33
2.2 - Créer des objets : choix d'un constructeur.....	34
2.2.1 - <i>Constructeur par défaut</i> .....	34
2.2.2 - <i>Définir son constructeur</i> .....	35
2.2.3 - <i>Allocation dynamique</i> .....	38
2.2.4 - <i>Rétablir le constructeur par défaut</i> .....	39
2.2.5 - <i>Plusieurs constructeurs dans une même classe</i> .....	40
2.2.6 - <i>Signature d'une fonction, arguments par défaut</i> .....	43
2.2.7 - <i>Fonctions à nombre variable d'arguments</i> .....	44

2.3 -	Macros et fonctions en ligne .....	47
2.4 -	Destructeur par défaut et destructeur explicite .....	50
2.4.1 -	<i>Le mécanisme implicite de destruction</i> .....	50
2.4.2 -	<i>Destructeur explicite</i> .....	51
2.4.3 -	<i>Les opérateurs new et delete</i> .....	52
2.4.4 -	<i>Portée et durée de vie d'une variable</i> .....	53
2.4.5 -	<i>Durée de vie d'une variable dynamique</i> .....	54
2.5 -	Fonctions : transmission de paramètres .....	55
2.5.1 -	<i>Echanger les valeurs de deux paramètres dans une fonction</i> .....	56
2.5.2 -	<i>Modes de transmission de paramètres en C++</i> .....	57
2.5.3 -	<i>&amp;, la spécification de référence</i> .....	58
2.5.4 -	<i>Valeurs renvoyées par une fonction (ou un opérateur)</i> .....	58
2.6 -	Et si nous reparlions des constructeurs ? .....	61
2.6.1 -	<i>Constructeur et conversion de type</i> .....	61
2.6.2 -	<i>Constructeur-copie</i> .....	63
2.6.3 -	<i>Importance du constructeur par défaut</i> .....	65
2.7 -	Le modificateur const.....	66
2.7.1 -	<i>Déclarer une constante dans un programme</i> .....	67
2.7.2 -	<i>Partager une constante entre plusieurs fichiers sources</i> .....	67
2.7.3 -	<i>Transmettre une constante en paramètre de fonction</i> .....	67
2.7.4 -	<i>Prévenir toute modification d'un paramètre dans une fonction</i> .....	68
2.7.5 -	<i>Pointeurs et constantes</i> .....	68
<b>3</b>	<b>Créer sa première classe fonctionnelle</b> .....	<b>71</b>
3.1 -	Une classe pour gérer les chaînes de caractères.....	71
3.2 -	Première implémentation de la classe String.....	73
3.2.1 -	<i>Les données membres</i> .....	73
3.2.2 -	<i>Définition de la classe String</i> .....	74
3.2.3 -	<i>Définition d'accessseurs pour la classe String</i> .....	76
3.3 -	Surcharge des opérateurs.....	78
3.3.1 -	<i>Désigner l'objet qui reçoit le message</i> .....	80
3.3.2 -	<i>Surcharge de l'opérateur d'indexation</i> .....	80
3.3.3 -	<i>Surcharge de l'opérateur d'affectation</i> .....	82
3.3.4 -	<i>Surcharge de l'opérateur +</i> .....	84
3.3.5 -	<i>Surcharge des opérateurs de comparaison</i> .....	86
3.4 -	Opérateurs d'entrées/sorties .....	88
3.4.1 -	<i>Surcharge de l'opérateur de sortie</i> .....	88
3.4.2 -	<i>Surcharge de l'opérateur d'entrée</i> .....	90
3.5 -	Compter le nombre d'instances d'une classe .....	91
3.5.1 -	<i>Variables de classe</i> .....	91
3.5.2 -	<i>Notion de méthode de classe</i> .....	93
3.5.3 -	<i>Utilisation des variables de classe pour la mise au point de la classe</i> .....	94
3.5.4 -	<i>Une classe String fonctionnelle</i> .....	95
3.6 -	Annexe : la classe String.....	96
3.6.1 -	<i>Fichier d'en-tête entete.h</i> .....	96
3.6.2 -	<i>Fichier d'en-tête c_String.h</i> .....	96
3.6.3 -	<i>Fichier c_String.cpp</i> .....	98

<b>4</b>	<b>Utiliser une classe existante : l'héritage</b> .....	103
4.1 -	Héritage : construire et utiliser la classe de base .....	103
4.1.1 -	<i>Une classe de base : Produit</i> .....	104
4.1.2 -	<i>Un premier exemple d'héritage : ProduitPerissable</i> .....	105
4.1.3 -	<i>Problèmes d'accès aux membres de la classe de base</i> .....	108
4.1.4 -	<i>Surcharge d'une donnée membre</i> .....	114
4.2 -	Héritage des fonctions membres et des opérateurs .....	115
4.2.1 -	<i>Constructeurs</i> .....	115
4.2.2 -	<i>Les fonctions virtuelles</i> .....	118
4.2.3 -	<i>Héritage des opérateurs</i> .....	123
4.2.4 -	<i>Transtypage et héritage</i> .....	128
4.3 -	Héritage multiple : ProduitFugace .....	130
4.3.1 -	<i>Créer une classe héritant de deux classes dérivées</i> .....	131
4.3.2 -	<i>Constructeur d'une classe à héritage multiple</i> .....	132
4.3.3 -	<i>Transtypage et classe de base virtuelle</i> .....	134
4.3.4 -	<i>Destructeurs virtuels</i> .....	136
<b>5</b>	<b>Construire et organiser une librairie de classes</b> .....	139
5.1 -	Une classe Tableau ? .....	139
5.1.1 -	<i>Des tableaux d'entiers</i> .....	139
5.1.2 -	<i>Un tableau de pointeurs</i> .....	141
5.2 -	Une classe Object et sa descendance .....	142
5.2.1 -	<i>Deux sous-classes pour la classe Object</i> .....	144
5.2.2 -	<i>Un Integer peut-il être un int ?</i> .....	146
5.2.3 -	<i>Comparer des objets comparables</i> .....	146
5.2.4 -	<i>Conversion par constructeur ou par cast ?</i> .....	149
5.2.5 -	<i>Le transtypage vers une référence</i> .....	150
5.2.6 -	<i>Comparer des objets non comparables ?</i> .....	150
5.3 -	Une classe Array pour répertorier des descendants de Object .....	154
5.3.1 -	<i>Répertorier un objet dans un tableau</i> .....	155
5.3.2 -	<i>Un objet pour représenter l'absence d'objet</i> .....	157
5.3.3 -	<i>Récapitulons les accesseurs de Array</i> .....	160
5.3.4 -	<i>Itérer sur le contenu d'une instance de Array</i> .....	162
5.3.5 -	<i>Une classe d'itérateurs</i> .....	163
5.4 -	Une hiérarchie de classes-conteneurs .....	166
5.4.1 -	<i>Une classe abstraite : Collection</i> .....	166
5.4.2 -	<i>La fonction membre NewIterator</i> .....	169
5.4.3 -	<i>Le polymorphisme de la fonction membre Includes</i> .....	171
5.4.4 -	<i>Pour terminer, une implémentation rapide</i> .....	172
<b>6</b>	<b>Développer une application en C++</b> .....	177
6.1 -	Description du distributeur .....	177
6.2 -	Les pièces et les produits .....	179
6.2.1 -	<i>Les pièces</i> .....	179
6.2.2 -	<i>Les produits</i> .....	181
6.2.3 -	<i>La classe Date : une classe annexe</i> .....	185
6.2.4 -	<i>Une amélioration de l'implémentation de la classe Object</i> .....	186
6.3 -	Le distributeur .....	187

6.3.1 - Une classe Sac .....	187
6.3.2 - Structure interne du distributeur .....	189
6.3.3 - Traitements disponibles .....	189
6.4 - Les clients.....	192
6.4.1 - Représentation du client .....	192
6.4.2 - Les actions du client .....	193
6.5 - Exemple d'exécution : la fonction main.....	194
6.6 - Annexe : les classes de l'application .....	198
6.6.1 - La hiérarchie de classes .....	198
6.6.2 - La classe Piece.....	198
6.6.3 - La classe Produit .....	200
6.6.4 - La classe Date.....	202
6.6.5 - La classe Sac.....	204
6.6.6 - La classe Distributeur.....	206
6.6.7 - La classe Client.....	212
<b>Annexe A : compléments C/C++ .....</b>	<b>217</b>
A1 - Séquences d'échappement .....	217
A2 - Mot-clé static.....	218
A3 - Arguments de la fonction main.....	220
A4 - Priorités des opérateurs .....	221
<b>Annexe B : les templates .....</b>	<b>223</b>
B1 - Pourquoi les templates ? .....	223
B2 - Une classe Bag en C++ .....	223
B3 - Définir une classe paramétrée .....	226
B4 - Modèles de fonctions .....	227
<b>Index .....</b>	<b>229</b>