

TABLE DES MATIÈRES

Avant-propos de la 2^e édition xvii

Avant-propos xix

Le langage C++

- 1 Introduction
 - 1.1 Le langage C 3
 - 1.2 Abstraction 3
 - 1.3 Programmation orientée objets 4
 - 1.4 Le langage C++ 4

- 2 Structure d'un programme C++
 - 2.1 Forme générale d'un programme C++ 7
 - Directives de compilation 8
 - Déclarations globales 8
 - Fonctions 8
 - 2.2 Symboles d'un programme C++ 10
 - Commentaires 10
 - Identificateurs 10
 - Mots-clés 11
 - Littéraux 12
 - 2.3 Exemple 13

- 3 Types de données
 - 3.1 Types fondamentaux 15
 - Types entiers 16
 - Types réels, à virgule flottante 16

	Type booléen	17
	Types caractères	17
	Type void	17
	Type énumération	18
3.2	Types dérivés	18
3.3	Types synonymes	19
4	Instructions simples	
4.1	Déclaration de constantes	21
4.2	Déclaration de variables	22
4.3	Affectation simple	23
4.4	Instructions d'entrées/sorties	23
	Entrées/sorties C++	23
	Entrées/sorties C	26
5	Expressions et opérateurs	
5.1	Opérateurs	29
	Opérateurs d'affectation	30
	Opérateurs arithmétiques	31
	Opérateurs relationnels	31
	Opérateurs logiques	32
	Incrémementation et décrémentation	32
	Opérateurs de décalage	33
	Opérateurs logiques bit-à-bit	34
	Opérateurs de transtypage	34
5.2	Ordre de précedence des opérateurs	35
5.3	Conversions de types	36
5.4	Fonctions mathématiques	38
6	Instructions de contrôle	
6.1	Séquence ou bloc	39
6.2	Sélections	40
	Instruction if	40
	Instruction switch	41
6.3	Itérations	42
	Instruction while	43
	Instruction do...while	43
	Instruction for	44
6.4	Sauts	46
	Instruction break	46
	Instruction continue	46
	Instruction return	47
	Instruction goto	47

7	Fonctions
7.1	Décomposition algorithmique 49
7.2	Déclaration d'une fonction 50
7.3	Définition d'une fonction 51
7.4	Appel d'une fonction 52
7.5	Variables locales et variables globales 53
7.6	Passage de paramètres 55
	Passage par valeur 55
	Passage par référence 56
7.7	Fonctions inline 57
7.8	Surcharge de fonctions 57
7.9	Type du résultat d'une fonction 58
8	Pointeurs et références
8.1	Pointeurs 61
	Déclaration de pointeurs 61
	Utilisation de pointeurs 62
	Pointeurs et constantes 63
8.2	Références 64
9	Tableaux
9.1	Notion de tableau 67
9.2	Tableaux unidimensionnels 68
	Déclaration 68
	Initialisation 68
	Accès aux éléments du tableau 69
9.3	Tableaux multidimensionnels 69
	Déclaration 69
	Initialisation 70
	Accès aux éléments du tableau 70
9.4	Tableaux et pointeurs 70
9.5	Paramètres tableaux 71
9.6	Opérateur sizeof 73
10	Chaînes de caractères
10.1	Nature d'une chaîne de caractères 75
	Littéraux chaînes de caractères 76
	Variables chaînes de caractères 76
	Pointeurs sur chaînes de caractères 77
10.2	Opérations sur les chaînes 78
	Lecture d'une chaîne 78
	Affichage d'une chaîne 78

	Copie de chaînes 79
	Comparaison de chaînes 79
	Longueur d'une chaîne 79
10.3	Fonctions de la librairie <cstring> 80
11	Structures et unions
11.1	Structures 81
	Déclaration de structures 81
	Accès aux membres d'une structure 83
	Utilisation des structures 83
11.2	Champs de bits 84
	Déclaration de champs de bits 84
	Utilisation de champs de bits 85
11.3	Unions 86
	Déclaration d'une union 86
	Utilisation d'unions 86
12	Allocation dynamique
12.1	Allocation dynamique de mémoire 89
12.2	Création d'un objet dynamique 90
12.3	Destruction d'un objet dynamique 91
12.4	Vérification de l'espace alloué 92
12.5	Allocation dynamique en C 93
12.6	Exemples 94
13	Espaces de noms et classes de stockage
13.1	Définitions 100
	Portée 100
	Durée de vie 101
	Emplacement mémoire 101
13.2	Espaces de noms 102
	Définition d'un espace de noms 103
	Utilisation d'un espace de noms 104
13.3	Spécificateurs de classes de stockage 106
	Spécificateur extern 106
	Spécificateur auto 106
	Spécificateur static 107
	Spécificateur register 108
	Spécificateur mutable 108
13.4	Exemples 109
14	Directives de prétraitement
14.1	Forme d'une directive 117
14.2	#define (définition) 118
	Définition de constantes 118
	Définition de macros 119

- 14.3 #undef (suppression de définition) 120
- 14.4 #include (inclusion de fichier) 120
- 14.5 Compilation conditionnelle 121
- 14.6 #line (contrôle des lignes) 123
- 14.7 #error (contrôle du message d'erreur) 124
- 14.8 #pragma 124

Programmation orientée objets en C++

- 15 L'orientation objets
 - 15.1 Qualités du logiciel 127
 - 15.2 Modèle à objets 128
 - Abstraction 129
 - Encapsulation 129
 - Modularité 130
 - Hierarchie 130
 - 15.3 Applications orientées objets 130
 - 15.4 Concepts de classe et d'objet 131
 - 15.5 Exemple 131

- 16 Classes et objets
 - 16.1 Exemple 133
 - 16.2 Abstraction de données et classes 134
 - Qu'est-ce qu'un type ? 135
 - Qu'est-ce qu'une classe ? 135
 - Encapsulation de membres 136
 - Définition des membres fonctions 137
 - Déclaration et utilisation d'objets 141
 - 16.3 Constructeurs et destructeurs 144
 - Constructeur par défaut 145
 - Constructeur avec paramètres 146
 - Destructeur 148
 - 16.4 Décomposition en fichiers 149
 - Séparer déclaration et définition 149
 - Exemple complet 150
 - 16.5 Pointeur this 153
 - 16.6 Tableaux d'objets 156
 - 16.7 Portée de classe 157

- 17 Compléments sur les classes
 - 17.1 Fonctions et classes amies 159
 - Fonctions amies 160
 - Classes amies 163

17.2	Classes à double représentation	163
17.3	Surcharge d'opérateurs	166
	Exemple de surcharge de l'opérateur +	167
	Multiplés surcharges d'un opérateur	169
	Surcharge d'opérateurs et fonctions amies	171
	Classe Vecteur : exemple complet	173
	Opérateurs de conversion	176
17.4	Encapsulation à base de classes	177
17.5	Membres statiques	178
	Attributs de classe statiques	178
	Méthodes de classe statiques	178
17.6	Classes et allocation dynamique	179
	Objets dynamiques	179
	Attributs pointeurs	180
17.7	Copie d'objets	181
	Constructeur de copie	181
	Affectation entre objets	183
17.8	Membres objets	184
18	Héritage et polymorphisme	
18.1	Classes dérivées	187
	Hiérarchie de classes	188
	Exemple	189
	Accès aux membres	191
	Constructeurs de classes dérivées	192
	Redéfinition de membres	193
	Compatibilité entre objets	193
	Destructeur de classes dérivées	195
	Membres protégés	195
	Types de dérivation	196
18.2	Polymorphisme	197
	Définition	197
	Liaisons statiques et dynamiques	197
	Fonctions virtuelles	198
	Classes abstraites	201
18.3	Héritage multiple	201
	Définition	201
	Classes virtuelles	203
19	Gestion des exceptions	
19.1	Principe de la gestion des exceptions	205
19.2	Structure de contrôle throw - try - catch	206
	throw	206
	try	207
	catch	208

19.3	Techniques de gestion d'exceptions 211
	Lancement d'objets 211
	Techniques de réception d'objets lancés 212
	Imbrication 213
	Relancement d'une exception 214
19.4	Spécifications d'exceptions 215
19.5	Exemple complet 215
19.6	Exceptions standard 218
20	Modèles de classes
20.1	Modèle de classe 221
	Déclaration d'un modèle 222
	Définition d'un modèle 223
	Utilisation d'un modèle 225
20.2	Classe conteneur 227
21	La Bibliothèque standard C++
21.1	Bibliothèque standard C++ 229
21.2	STL 230
21.3	Conteneurs 231
	Conteneurs séquentiels (séquences) 231
	Conteneurs associatifs 235
	Méthodes des conteneurs 240
	Adapteurs de conteneurs séquentiels 242
21.4	Itérateurs 243
	Fonctionnement des itérateurs 244
	Exemple 246
21.5	Algorithmes 247
	Classes d'algorithmes 247
	Exemples d'application d'algorithmes 248
21.6	Chaînes de caractères 251
	Construction de chaînes de caractères 251
	Taille d'une chaîne de caractères 252
	Manipulation de chaînes de caractères 252
	Recherche dans des chaînes de caractères 254
	Opérations sur les chaînes de caractères 254

Modélisation objets avec UML

22	Introduction au génie logiciel
22.1	Qu'est-ce que le génie logiciel? 259
22.2	Cycle de vie du logiciel 260

- 22.3 Cycles de développement du logiciel 261
 - Le modèle de la cascade 261
 - Le modèle de la spirale 262
- 22.4 Catégories de méthodes d'analyse et de conception 263
- 23 Approche objets
 - 23.1 Bases de l'approche objets 265
 - 23.2 Analyse et conception orientées objets 266
 - 23.3 Méthodes objets 267
- 24 UML
 - 24.1 Historique 269
 - 24.2 La notation UML 270
 - Diagrammes structurels 271
 - Diagrammes de comportement 273
 - Diagrammes de mise en œuvre 277
- 25 Etude de cas
 - 25.1 Bases de l'approche simplifiée 279
 - 25.2 Enoncé du problème *TicTacToe* 282
 - 25.3 Etude de faisabilité 283
 - Que détermine l'étude de faisabilité? 283
 - Etude de faisabilité pour *TicTacToe* 283
 - 25.4 Phase d'analyse (OOA) 284
 - Produits désirés de la phase d'analyse 284
 - Cas d'utilisation 285
 - Démarche 285
 - Résultats de la phase d'analyse 287
 - Personnes impliquées 288
 - Phase d'analyse pour *TicTacToe* 288
 - Conclusions de la phase d'analyse 296
 - 25.5 Phase de conception (OOD) 296
 - Produits désirés de la phase de conception 296
 - Modèles conceptuels 297
 - Stratégies pour la gestion des erreurs 299
 - Démarche 299
 - Résultats de la phase de conception 300
 - Personnes impliquées 301
 - Phase de conception pour *TicTacToe* 301
 - Conclusions de la phase de conception 308
 - 25.6 Phase d'implémentation (OOP) 308
 - Produits désirés de la phase d'implémentation 308
 - Démarche 309
 - Résultats de la phase d'implémentation 309
 - Personnes impliquées 310
 - Phase d'implémentation pour *TicTacToe* 310

- 25.7 Phase de test 319
 - Produits désirés de la phase de test 319
 - Techniques de test 320
 - Démarche 320
 - Résultats de la phase de test 321
 - Personnes impliquées 321
 - Phase de test pour *TicTacToe* 321
 - Conclusions de la phase de test 325
- 25.8 Conclusions du cycle de développement 325

Annexes 327

- A Structogrammes
- B Flots
 - B.1 Flots 334
 - Flots prédéfinis 334
 - Classe *istream* 334
 - Classe *ostream* 335
 - Surdéfinition de << et >> pour classes 335
 - Statut d'erreur d'un flot 336
 - Formatage 337
 - B.2 Association d'un flot à un fichier 337
- C La notation UML
 - C.1 Diagrammes structurels 339
 - Diagrammes de classes 339
 - Relations entre classes 340
 - Paquetages 340
 - C.2 Diagrammes de comportement 341
 - Diagrammes de cas d'utilisation 341
 - Diagrammes d'interaction 341
 - Diagrammes d'états-transitions 342
 - Diagrammes d'activités 343
 - C.3 Diagrammes de mise en œuvre 343
 - Diagrammes de composants 343
 - Diagrammes de déploiement 344
- D Liste des exemples 345
- Bibliographie 349
- Index 353

