

La bibliothèque standard STL du C++

Alain Bernard Fontaine

INFORMATIQUES

InterEditions



TABLE DES MATIERES

1. INTRODUCTION	1
1.1 Introduction	1
1.2 Structure de la bibliothèque	5
1.3 Versions	5
1.4 Livraison	6
1.5 Copyright	6
2. APPRENTISSAGE.....	7
2.1 Exemples progressifs	7
2.2 Définitions préliminaires	13
3. COMPLÉMENTS SUR LE C++	15
3.1 Modèles	16
3.2 Flots de données : streams	19
3.2.1 Hiérarchie des streams.....	20
3.2.2 Gestion de fichiers	23
3.3 Surcharge des opérateurs	24
3.4 Exceptions	26
3.4.1 Déclaration des exceptions dans les fonctions.....	28
3.4.2 L'en-tête exception.....	29
3.5 Mot clé « explicit »	33
3.6 Espaces de noms.....	34
4. ORGANISATION DE LA BIBLIOTHÈQUE C++	35
4.1 Les fonctions de la bibliothèque standard.....	35
4.2 La classe <i>string</i>	38
5. GESTION DE LA MÉMOIRE.....	41
5.1 Le problème.....	41
5.2 Déclarations	42
5.3 API des allocateurs	43
5.4 Fonctions d'allocation de l'en-tête <i>new</i>	44
5.5 Pointeurs automatiques	46
6. CONTENEURS.....	48
6.1 Définition	48
6.2 Gestion de mémoire	49
6.3 <i>typedef</i> communs aux conteneurs.....	49
6.4 Fonctions membres communes	50
6.5 Fonctions de construction	51
6.6 Fonctions d'insertion	51
6.7 Fonctions d'effacement.....	52
7. VECTEURS.....	53
7.1 Vecteurs	53
7.2 Définition	53
7.3 Création et destruction.....	53
7.4 Fonctions d'information	57

7.5 Fonctions de récupération des données.....	57
7.6 Fonctions d'insertion	58
7.7 Itérateurs	60
7.8 Destruction de cellules.....	61
7.9 Opérateurs.....	63
7.10 Implémentation du conteneur <i>vector</i>	64
7.11 Cas des booléens.....	67
8. LISTES.....	68
8.1 Listes.....	68
8.2 Définition	68
8.3 Création et destruction.....	68
8.4 Fonctions d'information	70
8.5 Fonctions de récupération des données.....	70
8.6 Fonctions d'insertion	71
8.7 Destruction	72
8.8 Itérateurs	73
8.9 Opérations sur les listes	74
8.10 Opérateurs	77
8.11 Implémentation.....	77
9. LISTES À DOUBLE EXTRÉMITÉ.....	79
9.1 Listes à double extrémité : deque	79
9.2 Définition	79
9.3 Création et destruction.....	80
9.4 Fonctions d'information	82
9.5 Fonctions de récupération des données.....	83
9.6 Fonctions d'insertion	83
9.7 Itérateurs	85
9.8 Destruction de cellules.....	85
9.9 Opérateurs	86
9.10 Implémentation.....	86
10. CONTENEURS ASSOCIATIFS.....	88
10.1 Définition	88
10.1.1 Itérateurs	89
10.1.2 Nouveaux typedef.....	90
10.2 Tables de correspondances : map	90
10.2.1 Définition.....	90
10.2.2 Création et destruction.....	91
10.2.3 Fonctions d'information.....	93
10.2.4 Accès aux données	93
10.2.5 Modification, insertion, destruction.....	94
10.2.6 Opérateurs sur les map	96
10.3 multimap	97
10.4 Implémentation.....	98
10.5 Ensemble : set.....	100
10.5.1 Définition.....	101
10.5.2 Création et destruction.....	101
10.5.3 Accès aux données	102
10.5.4 Insertion, destruction	103
10.5.5 Opérateurs sur les set.....	104
10.6 Ensemble multiple : multiset.....	104
10.6.1 Insertion, destruction	104

10.6.2 Implémentation	105
10.7 Remarques sur les fonctions membres modèles	105
11. ADAPTATEURS DE CONTENEURS	106
11.1 Définition	106
11.2 Pile : stack	107
11.2.1 Définition.....	108
11.2.2 Construction	108
11.2.3 Interface des fonctions : API	108
11.2.4 Implémentation	109
11.2.5 Exemples.....	109
11.3 File d'attente : queue	110
11.3.1 Déclaration.....	110
11.3.2 Constructeurs.....	111
11.3.3 Interface des fonctions : API	111
11.3.4 Opérations	111
11.4 File d'attente avec priorités : priority_queue	112
11.4.1 Déclaration.....	112
11.4.2 Constructeurs.....	113
11.4.3 Interface des fonctions : API	113
11.4.4 Opérations	114
12. OBJETS FONCTIONS	116
12.1 Définition	116
12.2 Fonctions arithmétiques	118
12.3 Fonctions de comparaisons	119
12.4 Fonctions logiques	121
12.5 Inversion logiques de fonctions	121
12.6 Relieurs : binders	122
12.7 Pointeurs vers des fonctions	123
13. FONCTIONS D'AIDE	125
13.1 Utilisation des <i>typedef</i>	125
13.2 Fonctions enveloppes	126
14. ITÉRATEURS.....	128
14.1 Définition	128
14.1.1 <i>typedef</i> pour les itérateurs.....	131
14.1.2 Temps de traversée.....	132
14.2 Classement des itérateurs	132
14.3 Itérateurs « entrée »	134
14.4 Itérateurs « sortie »	135
14.5 Itérateurs unidirectionnels.....	136
14.6 Itérateurs bidirectionnels.....	136
14.7 Itérateurs en accès direct	136
14.8 Types constants ou mutables	138
14.9 Associations entre itérateurs et conteneurs	139
15. ADAPTATEURS D'ITÉRATEURS	140
15.1 Adaptateurs d'inversion de sens de parcours	140
15.2 Adaptateurs d'insertion.....	141
15.3 Adaptateurs d'entrées-sorties	143
15.3.1 Itérateurs en flot d'entrée.....	143
15.3.2 Itérateurs en flot de sortie	145

16. ALGORITHMES	146
16.1 Introduction	146
16.2 Performances	148
16.3 Algorithmes non mutants	148
16.3.1 Application pour chaque valeur : <i>for_each</i>	149
16.3.2 Recherche : <i>find</i>	150
16.3.3 Recherche des valeurs adjacentes égales : <i>adjacent find</i>	152
16.3.4 Comptage d'une condition : <i>count</i>	153
16.3.5 Détection des différences : <i>mismatch</i>	154
16.3.6 Recherche de l'égalité : <i>equal</i>	156
16.3.7 Recherche de séquences : <i>search</i>	157
16.4 Algorithmes mutants	159
16.4.1 Recopie : <i>copy</i>	160
16.4.2 Echange : <i>swap</i>	161
16.4.3 Transformation d'un conteneur : <i>transform</i>	162
16.4.4 Remplacement : <i>replace</i>	163
16.4.5 Remplissage : <i>fill</i>	164
16.4.6 Production de valeurs selon une fonction : <i>generate</i>	165
16.4.7 Suppression sélective : <i>remove</i>	166
16.4.8 Elimination des valeurs multiples : <i>unique</i>	166
16.4.9 Inversion de l'ordre des données : <i>reverse</i>	168
16.4.10 Rotation circulaire des données : <i>rotate</i>	168
16.4.11 Distribution uniforme : <i>random_shuffle</i>	170
16.4.12 Partition d'un conteneur : <i>partition</i>	171
16.5 Algorithmes de tri et de fusion	172
16.5.1 Tris	173
16.5.2 Placement en nième position : <i>nth_element</i>	176
16.5.3 Recherche binaire : <i>binary search</i>	177
16.5.4 Bornes : <i>lower_bound</i> , <i>upper_bound</i> , <i>equal_range</i>	178
16.5.5 Fusion : <i>merge</i>	180
16.5.6 Opérations ensemblistes sur des structures triées	182
16.5.7 Opérations sur les « tas »	186
16.5.8 Minimum et maximum : <i>min</i> , <i>max</i>	189
16.5.9 Valeur médiane de trois valeurs : <i>median</i>	190
16.5.10 Comparaison lexicographique : <i>lexicographical_compare</i>	190
16.5.11 Permutations de séquences : <i>next(prev)_permutation</i>	191
16.6 Algorithmes généraux numériques	192
16.6.1 Accumulation de données : <i>accumulate</i>	192
16.6.2 Produit intérieur : <i>inner_product</i>	193
16.6.3 Somme partielle : <i>partial_sum</i>	194
16.6.4 Différence adjacente : <i>adjacent_difference</i>	194
17. EXEMPLE FINAL	196
ANNEXES	201
BIBLIOGRAPHIE	204
INDEX	206