



Norme
finale
ANSI/ISO



*R*essources D'EXPERTS

Jesse Liberty

- Description détaillée des fonctionnalités du C++
- Stratégies avancées d'analyse et de conception orientées objet
- Problèmes d'implémentation
- Classes conteneurs de la STL, itérateurs et algorithmes
- Structures de données dynamiques
- Techniques d'analyse et de hachage



Tous les codes sources sur
www.campuspress.fr



CAMPUSPRESS

2-005-451-1

2-005-451-1

C++

RESSOURCES D'EXPERTS

Jesse Liberty, et al.



CAMPUSPRESS
FRANCE

Retrouvez
tous nos livres sur
www.campuspress.fr

Table des matières

| | | | |
|--|----|--|-----|
| Introduction | 1 | Traduction des diagrammes de transition d'état en langage C++ | 73 |
| Structure de l'ouvrage | 2 | Traduction des diagrammes d'activités en langage C++ | 76 |
| Niveau requis | 3 | Résumé | 80 |
| Les logiciels nécessaires | 3 | Chapitre 3. Héritage, polymorphisme et réutilisation du code | 81 |
| Comment aborder ce livre | 4 | Avantages de l'héritage | 82 |
| Partie I. Programmation orientée objet .. | 5 | Les listes chaînées orientées objet | 83 |
| Chapitre 1. Conception et analyse orientées objet | 7 | Concevoir la liste chaînée | 83 |
| Construction des modèles | 8 | Implémenter la liste chaînée | 85 |
| Conception du logiciel : le langage de modélisation | 9 | Les classes abstraites | 91 |
| Conception du logiciel : le processus | 11 | Remplacer les méthodes virtuelles pures . | 96 |
| La vision | 13 | Les destructeurs virtuels | 97 |
| Analyse du cahier des charges | 14 | Polymorphisme via la surcharge des méthodes | 97 |
| Les cas d'utilisation | 14 | Gestion de la mémoire | 101 |
| Analyse de l'application | 26 | Surcharge des autres opérateurs | 106 |
| Analyse des systèmes | 26 | L'opérateur d'affectation | 107 |
| Les documents de planification | 27 | Surcharge des opérateurs d'incrémentat | 110 |
| Les visualisations | 28 | Les constructeurs de copie virtuels | 114 |
| Les documents de synthèse | 28 | Héritage multiple | 115 |
| Conception | 29 | Les problèmes de l'héritage multiple | 116 |
| A quoi correspondent les classes ? | 29 | Héritage multiple contre relation contenant-contenu | 121 |
| Les transformations | 31 | Résumé | 123 |
| Le modèle statique | 33 | Partie II. Les problèmes d'implémentation | 125 |
| Le modèle dynamique | 43 | Chapitre 4. Gestion de la mémoire | 127 |
| Résumé | 47 | Gestion de la mémoire et pointeurs | 129 |
| Chapitre 2. Implémentation de la conception des classes en C++ | 49 | Les fuites de mémoire | 132 |
| Traduction des diagrammes de classes en langage C++ | 50 | Les pointeurs "errants" | 134 |
| Les classes standards | 50 | Les pointeurs <i>const</i> | 135 |
| Les classes modèles | 52 | Les pointeurs <i>const</i> et les fonctions membres <i>const</i> | 136 |
| Les classes utilitaires | 53 | Transmission par référence | 136 |
| Les associations | 54 | Transmission d'un pointeur <i>const</i> | 137 |
| Les agrégations | 61 | Ne pas renvoyer la référence d'un objet qui ne se trouve pas dans la portée | 137 |
| La généralisation | 63 | Qui détient le pointeur ? | 141 |
| Traduction des diagrammes d'interactions en langage C++ | 65 | | |
| Implémentation des diagrammes de collabo- ration et des diagrammes de séquence en C++ | 66 | | |

| | | | |
|--|-----|--|-----|
| Pointeurs et exceptions | 141 | La bibliothèque C++ standard | 277 |
| Utilisation des pointeurs auto | 146 | Conception des types d'éléments | 278 |
| Comptabilisation des références | 152 | Résumé | 279 |
| Résumé | 166 | Chapitre 7. Itérateurs et algorithmes | |
| Chapitre 5. Comment utiliser les structures | 167 | de la STL | 281 |
| Les Microsoft Foundation Classes | 168 | Les classes d'itérateurs | 282 |
| Comment démarrer | 168 | La position dans un conteneur | 282 |
| Les autres assistants | 172 | Les types d'itérateurs en entrée et en sortie | |
| Vue d'ensemble | 172 | des conteneurs | 282 |
| L'architecture de l'application | 173 | La classe d'itérateur de base | 283 |
| Les multithreads | 173 | Les itérateurs d'entrée | 285 |
| Le multithread coopératif versus préventif | 174 | Les itérateurs de sortie | 286 |
| Multithread préventif | 175 | Les itérateurs en-avant | 286 |
| Une étude de cas simple | 176 | Les itérateurs bidirectionnels | 287 |
| La création des threads | 177 | Les itérateurs à accès aléatoire | 287 |
| Les classes utilitaires | 197 | Les opérations sur les itérateurs | 287 |
| Les classes de manipulation de chaînes .. | 197 | Les classes d'itérateurs standards | 289 |
| Les classes de temps | 197 | Objets fonctions | 292 |
| Documents et vues | 199 | Les prédicats | 293 |
| Les vues | 200 | Les fonctions arithmétiques | 294 |
| Résumé | 213 | Les classes d'algorithmes | 295 |
| Chapitre 6. Classes conteneur de | | Les opérations de séquence | |
| la bibliothèque des modèles standards | 215 | sans modification | 296 |
| Définition et instanciation des modèles | 216 | Les opérations de séquence | |
| Définir et instancier les modèles | | avec modification | 307 |
| de fonctions | 216 | Le tri et les opérations de séquence | |
| Définir et instancier les modèles de classes | 217 | associées | 321 |
| Présentation des conteneurs de séquence ... | 218 | Les compositions de fonctions standards . | 345 |
| Le conteneur vector | 218 | Résumé | 352 |
| Le conteneur list | 237 | Chapitre 8. Suppression des conflits | |
| Le conteneur deque | 249 | de noms à l'aide des espaces de noms | 353 |
| Présentation des piles | 251 | Les fonctions et les classes sont identifiées | |
| Présentation des files d'attente | 255 | par leur nom | 354 |
| Les files d'attente à priorité | 257 | Création d'un espace de noms | 358 |
| Présentation des conteneurs associatifs | 259 | Déclarer et définir les types | 359 |
| Le conteneur map | 259 | Définir des fonctions en dehors | |
| Le conteneur multimap | 272 | d'un espace de noms | 360 |
| Le conteneur set | 275 | Ajouter de nouveaux membres | 360 |
| Le conteneur multiset | 275 | Imbriquer les espaces de noms | 361 |
| Considération des performances | 276 | Utilisation d'un espace de noms | 361 |

| | | | |
|---|-----|---|-----|
| Le mot clé <i>using</i> | 365 | Avantages et inconvénients du RTTI | 414 |
| La directive <i>using</i> | 365 | Gestion de la mémoire pour les objets | |
| La déclaration <i>using</i> | 367 | temporaires | 417 |
| Les alias d'espace de noms | 369 | Résumé | 420 |
| Les espaces de noms non nommés | 369 | | |
| L'espace de noms standard <i>std</i> | 370 | Partie III. Manipulation des données | 421 |
| Résumé | 371 | | |
| Chapitre 9. Manipulation des types d'objets | | Chapitre 11. La récurrence et les structures | |
| à l'exécution | 373 | de données récurrentes | 423 |
| L'opérateur <i>typeid()</i> | 374 | Présentation de la récurrence | 424 |
| La classe <i>type_info</i> | 375 | Les nombres de Fibonacci : une définition | |
| Le constructeur de la classe <i>type_info</i> | 375 | récurrence | 424 |
| Les opérateurs de comparaison | 375 | Interrompre la récurrence | 426 |
| La fonction membre <i>name()</i> | 377 | Les structures récurrentes | 427 |
| La fonction membre <i>before()</i> | 380 | Parcours d'une structure récurrente avec | |
| L'opérateur <i>typeid()</i> dans les constructeurs | | une fonction récurrente | 429 |
| et les destructeurs | 381 | La récurrence versus itération et récurrence | |
| Les mauvaises utilisations de <i>typeid()</i> | 382 | fermée | 433 |
| Conversion de type explicite des objets .. | 384 | La récurrence fermée | 437 |
| L'opérateur <i>dynamic_cast()</i> | 384 | Récurrence indirecte | 438 |
| L'opérateur <i>typeid()</i> contre l'opérateur | | La récurrence et la pile | 439 |
| <i>dynamic_cast</i> | 390 | Débogage des fonctions récurrentes | 441 |
| Les autres opérateurs de conversion | 390 | Résumé | 441 |
| L'opérateur <i>static_cast()</i> | 390 | | |
| L'opérateur <i>reinterpret_cast()</i> | 392 | Chapitre 12. Conception de méthodes | |
| Quand utiliser l'opérateur <i>dynamic_cast()</i> , | | de tri efficaces | 443 |
| <i>static_cast()</i> ou <i>reinterpret_cast()</i> | 393 | Analyse des performances des algorithmes .. | 444 |
| L'opérateur <i>const_cast()</i> | 393 | Comparer la meilleure situation, la plus | |
| Comparaison des nouvelles et des anciennes | | défavorable et le cas moyen | 445 |
| conversions de type explicites | 395 | La stabilité des tris | 446 |
| Résumé | 396 | Utiliser de la mémoire supplémentaire | |
| Chapitre 10. Optimisation des performances | | pendant le tri | 447 |
| des applications | 397 | Le tri bubble | 447 |
| Les fonctions inline en dehors des définitions | | Analyse du tri bubble | 449 |
| de classe | 398 | Le tri par insertion | 450 |
| Ne pas laisser apparaître le code | | Analyse du tri par insertion | 452 |
| d'implémentation dans les fichiers | | Le tri par sélection | 452 |
| en-tête distribués | 404 | Analyse du tri par sélection | 455 |
| Analyse du coût des fonctions virtuelles | | Le tri rapide | 455 |
| et des classes de base virtuelles | 406 | Analyse du tri rapide | 458 |
| Les fonctions virtuelles | 406 | Le tri par fusion | 458 |
| Les classes de base virtuelles | 410 | Analyse du tri par fusion | 462 |

| | | | |
|--|------------|---|------------|
| Le tri shell | 463 | Analyse syntaxique | 525 |
| Analyse du tri shell | 465 | Analyse syntaxique des expressions | |
| Le tri sur le tas | 465 | numériques | 526 |
| Analyse du tri sur le tas | 468 | Analyse syntaxique des expressions chaînes | 528 |
| Choix d'une méthode de tri | 468 | Grammaire sans contexte et à analyse | |
| Génération de données de test | 470 | syntaxique | 530 |
| Résumé | 472 | Utilisation de l'analyse syntaxique progressive | |
| Chapitre 13. Les algorithmes de recherche | | pour valider les expressions régulières | 531 |
| en C++ | 473 | Résumé | 531 |
| Les recherches linéaires | 474 | Partie IV. Persistance des objets et | |
| Analyse de la recherche linéaire | 476 | chiffrement | 533 |
| La recherche sur correspondance | 479 | Chapitre 15. Persistance des objets | 535 |
| L'algorithme de la force brutale | 480 | Création d'objets stockables | 537 |
| Représentations du modèle | 481 | Qu'est-ce qu'un B-tree ? | 546 |
| Construire des machines à états finis | 481 | L'écriture sur disque | 549 |
| Les algorithmes sur les graphiques | 484 | Utilisation de la mémoire cache | 551 |
| La recherche d'abord-en-profondeur | 486 | Déterminer la taille de chaque page | 551 |
| La recherche par-niveau | 487 | Déterminer le nombre de pages | |
| Comparaison des recherches | | en mémoire à un instant donné | 552 |
| d'abord-en-profondeur et par-niveau | 488 | Echanges avec le disque | 552 |
| La recherche meilleur-d'abord | 489 | Implémentation du B-tree | 553 |
| Implémentation des objets graphiques | 491 | Mode de fonctionnement | 586 |
| Représentation du morpion | 495 | Déroulement du code | 588 |
| Appliquer des raccourcis alpha-bêta | 495 | La recherche | 604 |
| Le problème du voyageur de commerce | 497 | Résumé | 607 |
| La recherche externe | 499 | Chapitre 16. Bases de données relationnelles | |
| L'accès séquentiel indexé | 499 | et persistance | 609 |
| Les arbres binaires | 499 | Principes fondamentaux des bases de données | |
| Les arbres 2-3-4 | 500 | relationnelles | 610 |
| Résumé | 502 | Architecture d'une base de données | |
| Chapitre 14. Techniques de hachage et | | relationnelle | 613 |
| d'analyse syntaxique | 505 | Restrictions et considérations | 613 |
| Recherche versus hachage | 506 | SQL : définition et interrogation de la base | |
| Fonctions de hachage | 507 | de données | 614 |
| Résolution des collisions | 508 | Normalisation | 615 |
| Le rehachage linéaire | 508 | Les jointures | 616 |
| Le rehachage non linéaire | 509 | Persistance et base de données relationnelle | 617 |
| Densité d'occupation (alpha) | 510 | Les identificateurs d'objets | 617 |
| Chainages | 511 | Utiliser les blobs | 618 |
| Adressage par compartiment | 512 | | |

| | | | |
|---|------------|---|------------|
| Masquer les détails | 619 | Granularité du transfert des données | 721 |
| Stocker directement les objets | 619 | Granularité du verrouillage des données .. | 721 |
| Utiliser l'API de votre base de données .. | 619 | Résumé | 722 |
| Accéder aux sources de données ODBC .. | 620 | Chapitre 19. Protection des applications | |
| Les MFC | 623 | avec le chiffrement | 723 |
| Les instructions SQL | 637 | Historique de la cryptographie | 724 |
| Définir les caractéristiques de la base | | Le rôle du bureau national des standards .. | 725 |
| de données | 638 | Présentation du chiffrement | 726 |
| Résumé | 638 | Les codes | 727 |
| Chapitre 17. Persistance des objets à l'aide | | Les chiffrements | 728 |
| des bases de données relationnelles | 639 | Le chiffre de Gilbert Vernam | 730 |
| Les objets dans Oracle8 | 641 | Chiffrement à clé privée | 733 |
| Les types d'objets | 641 | Les algorithmes à clé privée | 734 |
| Les références d'objets | 642 | Le fonctionnement du chiffrement | |
| Les collections | 644 | à clé secrète | 736 |
| Les procédures externes développées en C++ | 646 | Les centres de distribution de clés | 739 |
| Association de diagrammes UML à une base | | Chiffrement à clé publique | 739 |
| de données relationnelle objet | 649 | La technique du puzzle de Ralph Merkle .. | 740 |
| Conception de la base de données | 650 | Les techniques de chiffrement | |
| Génération du code C++ | 651 | multi-utilisateur de Diffie-Hellman | 741 |
| Les générations du code de C++ | 652 | La technique RSA | 742 |
| Génération du serveur | 654 | Utilisation de PGP (<i>Pretty Good Privacy</i>) ... | 743 |
| Etude de cas : système de commande | 660 | Choix des nombres premiers dans PGP .. | 744 |
| Description du système | 660 | Utiliser des nombres aléatoires | |
| Chapitre 18. Les bases de données orientées | | en chiffrement | 745 |
| objet | 665 | Chiffrement de fichiers à l'aide de PGP .. | 746 |
| Présentation des SGBDO | 666 | Limites du chiffrement | 748 |
| Le standard ODMG | 668 | Les contraintes légales en France | 750 |
| Une application de facturation C++ | 668 | Attaques des fonctions de chiffrement | 752 |
| Persistance des données | 687 | Décrypter un fichier chiffré avec PGP ... | 753 |
| Schéma de base de données et outils | | Les signatures électroniques | 754 |
| de capture de schéma | 689 | Le standard de chiffrement à clé publique | |
| Les collections | 692 | (PKCS) | 755 |
| Les itérateurs | 695 | Le standard de signature électronique | |
| Les relations | 695 | (DSS) | 755 |
| Base de données et transactions | 702 | Non-refus | 756 |
| A propos des SGBDO | 719 | Les produits de chiffrement commercialisés .. | 757 |
| L'architecture client/serveur | 719 | Les clients e-mail sûrs | 758 |
| Stockage des données et regroupement | | Les produits de Bureau sécurisés | 759 |
| des objets | 719 | Résumé | 763 |

| | | | |
|--|-----|---|-----|
| Partie V. L'informatique distribuée | 765 | Performances | 795 |
| Chapitre 20. CORBA | 767 | Les fuites de mémoire des ORB | 795 |
| Théorie et justification | 769 | Granularité de l'interface | 795 |
| L'environnement CORBA minimal | 769 | Transmettre des références d'objets | 796 |
| Une structure pour la technologie objet | 771 | Résumé | 796 |
| IIOP : la colle des objets | 771 | Chapitre 21. COM | 799 |
| Un modèle de composant | 773 | Les principes fondamentaux de COM | 801 |
| IDL : le contrat de liaison | 773 | L'architecture COM | 801 |
| Comparaison de l'interface IDL avec une définition de classe C++ | 775 | Les interfaces | 802 |
| ORB (Object Request Broker) | 776 | L'interface IUnknown | 807 |
| Durée de vie de l'objet | 777 | Les objets COM | 811 |
| Les environnements de développement | 778 | Les bibliothèques de types | 814 |
| Comparaison des environnements CORBA | 779 | Les autres technologies COM | 815 |
| Interopérabilité des ORB | 779 | Utilisation des objets COM en C++ | 821 |
| Création du client C++ | 780 | Les interfaces brutes | 821 |
| Générer le stub | 781 | Les pointeurs intelligents | 822 |
| Se connecter à l'ORB | 782 | Les bibliothèques de types | 828 |
| Invoquer les méthodes | 782 | L'écriture des objets COM en C++ | 832 |
| L'application client C++ complète | 783 | L'héritage multiple | 832 |
| Création du serveur C++ | 784 | Les classes imbriquées | 835 |
| Générer le squelette | 784 | Les classes éphémères | 839 |
| Implémenter les méthodes du serveur | 785 | Résumé | 839 |
| Se connecter à la classe serveur | 786 | Bibliographie | 840 |
| Charger le BOA dans l'ORB | 786 | Chapitre 22. Java et C++ | 841 |
| Un client Java | 788 | Les similitudes entre C++ et Java | 842 |
| Générer le stub | 789 | Les commentaires | 842 |
| Code de démarrage et d'invocation des méthodes | 790 | Les types de données | 843 |
| Stratégies de test | 790 | Les opérateurs | 845 |
| Les traces | 791 | Les instructions de contrôle | 846 |
| Services d'enregistrement et de contrôle | 791 | Les différences entre C++ et Java | 846 |
| Gestion des exceptions | 791 | La gestion de la mémoire | 847 |
| Déboguer à distance | 792 | Pas de pointeur | 847 |
| Service Nommage et interopérabilité | 792 | Pas de préprocesseur | 847 |
| Référence d'objet interopérable (IOR ou Interoperable Object Reference) | 793 | Pas de destructeur | 847 |
| Les contextes de Nommage | 793 | Les spécificateurs d'accès | 848 |
| L'interopérabilité | 794 | Les paramètres de méthode | 849 |
| | | Les fonctions externes | 851 |
| | | Les énumérations | 851 |
| | | Les chaînes | 851 |
| | | Les tableaux | 852 |

| | | | |
|--|-----|---------------------------------|-----|
| Les fonctionnalités orientées objet de Java .. | 852 | La gestion des exceptions | 862 |
| Les classes | 853 | Résumé | 863 |
| Héritage | 859 | Index | 867 |
| L'héritage multiple | 862 | | |