

Sommaire

Partie A – Les enjeux

1. La qualité du logiciel
2. Critères d'orientation objet

Partie B – Vers l'objet

3. Modularité
4. Approches de la réutilisabilité
5. Vers la technologie objet
6. Types abstraits de données

Partie C – Techniques orientées objet

7. Structure statique : les classes
8. Structure d'exécution : les objets
9. Gestion de la mémoire
10. Généricité
11. Conception par contrat (assertions, pré/post-conditions)
12. Traitement des exceptions
13. Mécanismes complémentaires
14. Introduction à l'héritage
15. Héritage multiple
16. Techniques avancées d'héritage
17. Typage
18. Objets globaux et constantes

Partie D – Méthodologie

19. Sur la méthodologie
20. Design pattern : systèmes interactifs multi-écrans
21. Étude de cas héritage : "undo"
22. Comment identifier les classes
23. Principes de conception des classes
24. Bien utiliser l'héritage
25. Techniques utiles
26. Sens du style
27. Analyse orientée objet
28. Processus de construction logicielle
29. Enseigner la méthode

Partie E – Techniques avancées

30. Concurrence, distribution, client-serveur, Internet
31. Persistance d'objets et bases de données
32. Techniques OO pour interfaces graphiques

Partie F – Applications et langages

- 33. Programmation OO et Ada
- 34. Émulation OO en langages non OO
- 35. De Simula à Java : principaux langages OO

Partie G – Bien faire

- 36. Un environnement orienté objet
Épilogue : le langage dévoilé

Partie H – Annexes

- A. Extraits de la bibliothèque Base
- B. Généricité vs héritage
- C. Principes, règles, définitions
- D. Glossaire de la technologie objet
- E. Bibliographie (plus de 400 références)