Table des matières

Préface 1
Gilles DOWEK et Catherine DUBOIS

Avant-propos 3

Chapitre 1. Gestion des noms : modèle et opérations 5

1.1. Réutiliser, découper, confiner 6
1.1.1. Analyse des besoins 6
1.1.2. Répondre à ces besoins 7
1.2. Espaces de noms 9
1.2.1. Définition des espaces de noms 10
1.2.2. Extension de la notion d'environnement et de liaison 12
1.3. Développement des *kits* 16
1.3.1. Espace des noms d'un *kit* 18
1.3.2. Hypothèses d'étude 18
1.3.3. Type d'un *kit* 19
1.3.4. Valeur d'un *kit* 22
1.3.5. Exportation d'un *kit*, confinement de champs 25
1.3.6. Importation d'un *kit* 31
1.3.7. Étapes de développement 36
1.4. *Kits* incomplets 36
1.4.1. Type et valeur d'un *kit* incomplet 37
1.4.2. Complétion d'un *kit* incomplet 37
1.4.3. Confinement de champs d'un *kit* incomplet 38
1.5. *Kits* paramétrés 39
1.5.1. *Kits* paramétrés par un type 39
1.5.2. *Kits* paramétrés par un type et une valeur 43
1.5.3. Confinement, paramétrage, *kits* incomplets et exportation 47

1.6. Foncteurs de *kit* 48
1.7. Extension d'un *kit* 51
1.7.1. Présentation de l'extension 51
1.7.2. Confinement et extension 56
1.8. Commentaires sur le modèle de *kit* 61

Chapitre 2. Les modules 63

2.1. Les modules en Ada 64
2.1.1. Développement des modules 64
2.1.2. Exportation et confinement 67
2.1.3. Imbrication de modules 68
2.1.4. Importation d'un module 68
2.1.5. Étalement d'une importation 69
2.1.6. Modules génériques 70
2.1.7. Modules et compilation séparée 74
2.2. Les modules en OCaml 74
2.2.1. Définition d'un module 75
2.2.2. Exportation et confinement 76
2.2.3. Confinement des définitions de types 81
2.2.4. Foncteurs 85
2.3. Modularité, espaces de noms et *W-kit* 88
2.3.1. Interfaces de déclaration 89
2.3.2. *W-kit* 89
2.3.3. Modularité et fichiers d'en-tête en C 89

Chapitre 3. Traits orientés objet 95

3.1. Principaux traits objet 95
3.1.1. Objets 96
3.1.2. Classes 98
3.2. *Kits* et traits objet 106
3.2.1. Modéliser les classes 107
3.2.2. Modéliser les objets 109
3.2.3. Héritage, redéfinition et liaison retardée 112
3.2.4. *C-kits* incomplets, *C-kits* paramétrés 116
3.2.5. Héritage, sous-classage, sous-typage 117
3.2.6. Langages de types, classes et objets 121

Chapitre 4. Les classes dans différents langages 125

4.1. Les classes de Java 125
4.1.1. Présentation 125
4.1.2. Modules et packages 126
4.1.3. Classes 127

4.1.4. Marques . . . 133
4.1.5. Développement de classes . . . 136
4.2. Les classes de C++ . . . 146
4.2.1. Fichiers d'en-tête, espace de noms, confinement . . . 146
4.2.2. Un tour des classes . . . 150
4.2.3. Héritage et confinement . . . 156
4.2.4. Surcharge en C++ . . . 167
4.2.5. Classes paramétrées . . . 170
4.3. Les classes de OCaml . . . 176
4.3.1. Présentation . . . 176
4.3.2. Un tour des classes . . . 176
4.3.3. Marques, classes incomplètes, paramétrisation . . . 182
4.3.4. Objets . . . 186
4.3.5. Signatures de classe : confinement et héritage . . . 192
4.3.6. Héritage multiple . . . 195
4.3.7. Autres traits . . . 197
4.4. Les classes de Python . . . 198
4.4.1. Présentation . . . 198
4.4.2. Un tour des classes, modules et types . . . 199
4.4.3. Noms et affectation . . . 208
4.4.4. Affectation et typage . . . 212
4.4.5. Surcharge . . . 214
4.4.6. Modules et packages . . . 218
4.4.7. Confinement . . . 220
4.4.8. Héritage . . . 220
4.4.9. *C-kits* incomplets et classes abstraites . . . 223
4.4.10. Autres traits . . . 224

Annexe. Questions pour étudier un langage . . . 227

Liste des notations . . . 231

Bibliographie . . . 233

Index . . . 235

Sommaire de *Concepts et sémantique des langages de programmation 1* . . . 239