



Apprentissage de la programmation avec OCaml

Catherine Dubois, Valérie Ménéssier-Morain

Dubois, Catherine (1963-....)

Ménéssier-Morain, Valérie (1965-....)

Hermès science publications, Paris

ISBN: 2-7462-0819-9

Table des Matières

Apprentissage de la programmation avec OCaml

Catherine Dubois/Valérie Ménéssier-Morain

Hermes Science publications

Avant-propos	13
Chapitre 1. Introduction	17
1.1. Pourquoi étudier l'informatique (et en particulier la programmation) ?	17
1.2. Qu'est-ce-que la programmation ?	18
1.3. Les étapes du développement d'un programme	18
1.3.1. Première étape : la spécification	18
1.3.2. Deuxième étape : la recherche d'un algorithme pour résoudre le problème	19
1.3.3. Troisième étape : la transcription de l'algorithme dans un langage de programmation ou «écriture du programme»	19
1.3.4. Quatrième étape : la validation du programme	20
1.3.5. Récapitulatif	20
1.3.6. Exemple	21
1.4. Les langages de programmation	21
1.5. Pourquoi OCaml ?	23
1.6. Comment se procurer OCaml ?	23
1.7. Plan de l'ouvrage	23
1.8. Logiciels utilisés dans cet ouvrage	24
Première partie. Éléments de programmation	25
Chapitre 2. Introduction à OCaml	27
2.1. Notions d'expression et de valeur	27
2.2. Les types de base	28
2.2.1. Le type int	28
2.2.2. Le type float	28
2.2.3. Le type bool	29
2.2.4. Le type char	29
2.2.5. Le type string	30
2.3. La boucle interactive de OCaml ou comment utiliser OCaml ?	30
2.4. Définitions	33
2.4.1. Notion d'identificateur	33
2.4.2. Définitions globales	33
2.4.3. Définitions locales	37

2.5. Expressions conditionnelles	41
2.6. De l'usage des parenthèses	42
Chapitre 3. Fonctions	43
3.1. Fonctions d'une variable	43
3.1.1. Introduction	43
3.1.2. Définition de fonctions	44
3.1.3. Exemples	45
3.1.4. Application d'une fonction	46
3.2. Fonctions à plusieurs paramètres	47
3.2.1. Définition d'une fonction à plusieurs paramètres	47
3.2.2. Application d'une fonction à plusieurs paramètres	48
3.3. Composition de fonctions	49
3.4. Définition locale de fonction	50
3.5. Portée statique	51
3.6. Introduction au typage d'une fonction	52
3.7. Restriction du domaine de définition d'une fonction	54
3.8. Un exemple complet : écrire les nombres en lettres	55
3.8.1. Les fonctions unité, dizaine, centaine et millier	58
3.8.2. La fonction écrire_millier	59
3.8.3. La fonction écrire_centaine	60
3.8.4. La fonction écrire_dizaine_unité	60
Chapitre 4. Fonctions récursives	65
4.1. Notion de récursivité	66
4.2. Définition d'une fonction récursive en OCaml	67
4.2.1. Fonctions simplement récursives	67
4.2.2. Fonctions mutuellement récursives	69
4.3. Évaluation d'un appel à une fonction récursive	70
4.4. Propriétés des fonctions récursives	73
4.5. Méthode de décomposition récursive	75
4.5.1. Méthode générale	75
4.5.2. Un exemple simple : le calcul du reste de la division euclidienne de deux nombres naturels	76
4.5.3. Un exemple magique : les tours de Hanoi	76
Chapitre 5. Listes	79
5.1. Syntaxe - Typage - Évaluation	79
5.1.1. Définition	79
5.1.2. Syntaxe et évaluation	80
5.1.3. Typage	80
5.1.4. L'opérateur ::	81
5.2. Fonctions simples sur les listes	82
5.3. Filtrage	87
5.3.1. Vocabulaire et définitions	87
5.3.2. Récapitulatif des formes possibles de filtre	89

5.3.3. Mise en correspondance d'un filtre avec une valeur	90
5.3.4. Évaluation d'une expression de filtrage	90
5.4. Fonctions récursives sur les listes	91
5.4.1. Longueur d'une liste	91
5.4.2. Recherche d'un élément dans une liste	92
5.4.3. Dernier élément d'une liste	93
5.4.4. Recherche du plus petit élément d'une liste	94
5.4.5. Intervalle d'entiers	94
5.4.6. Extraction des entiers pairs d'une liste d'entiers	95
5.4.7. Remplacement de certains éléments	95
5.4.8. Concaténation de deux listes	96
5.5. Manipulation de listes triées	96
5.5.1. Recherche d'un élément dans une liste triée	97
5.5.2. Insertion dans une liste triée	97
5.6. Algorithmes de tri	99
5.6.1. Tri par insertion	99
5.6.2. Tri par sélection	100
5.6.3. Tri par fusion	100
5.6.4. Comparaison de ces différentes méthodes de tri	103
Chapitre 6. Types produits (paires et n-uplets)	107
6.1. Produit de deux types	107
6.2. Construction des paires (syntaxe, typage, évaluation)	108
6.3. Quelques exemples simples de fonctions manipulant des paires	108
6.4. Fonction récursive retournant une paire	111
6.5. Filtrage des couples	112
6.6. Listes et paires	112
6.6.1. Listes de vecteurs	113
6.6.2. Listes d'associations	113
6.7. Généralisation des couples : les n-uplets	115
6.8. Fonctions à plusieurs arguments versus fonctions à un argument n-uplet	116
6.9. Un exemple plus conséquent : la date du lendemain	117
Chapitre 7. Types enregistrements	121
7.1. Déclaration d'un type enregistrement	122
7.2. Création et manipulation d'enregistrements	124
7.3. Accès aux composantes d'un enregistrement	124
7.4. Filtrage des enregistrements	126
7.5. Un petit exemple : les notes d'un étudiant	127
7.6. Un exemple plus conséquent : les nombres rationnels	129
7.6.1. Représentation non normalisée (sans simplification systématique)	129
7.6.2. Simplifier les rationnels	130
7.6.3. Représentation normalisée	131
7.6.4. Simplifier en calculant	132

Chapitre 8. Types sommes	135
8.1. Exemple de la définition du type des formes géométriques	135
8.2. Déclaration d'un type somme	137
8.3. Construction des expressions d'un type somme (syntaxe, typage, évaluation)	138
8.4. Filtrage sur les types sommes	139
8.4.1. Manipulations de formes	139
8.4.2. Bilan sur les filtres	141
8.5. Modélisation d'un jeu de tarot	142
8.5.1. Définition des cartes ordinaires	142
8.5.2. Définition privilégiant la nature des cartes	143
8.5.3. Définition préservant la symétrie nature/couleur	144
8.5.4. Définition du type des cartes	144
Chapitre 9. Types récursifs	147
9.1. Polynômes à une indéterminée	147
9.2. Cocktails	152
9.3. Expressions arithmétiques	156
9.4. Arbres	159
9.5. Arbres binaires de recherche	164
9.5.1. Présentation	164
9.5.2. Recherche dans un arbre binaire de recherche	165
9.5.3. Parcours d'un arbre binaire de recherche	167
9.5.4. Insertion dans un arbre binaire de recherche	168
9.5.5. Construction d'un arbre binaire de recherche	168
9.6. Types mutuellement récursifs	169
Chapitre 10. Introduction à l'ordre supérieur	171
10.1. Fonctions numériques	172
10.1.1. Calcul de la pente d'une fonction en 0	172
10.1.2. Dérivée d'une fonction	172
10.1.3. Calcul d'un zéro d'une fonction par une méthode dichotomique	174
10.1.4. Composition de fonctions	176
10.2. Itération d'une fonction	177
10.2.1. Calcul du n-ième terme d'une suite définie récursivement	177
10.2.2. Sommation	178
10.2.3. Application d'une fonction à tous les éléments d'une liste	179
10.2.4. Itérer une fonction sur une liste	180
10.3. Fonctions paramétrées par un prédicat	184
10.3.1. Extraction d'éléments dans une liste	184
10.3.2. Insertion dans une liste triée, tri d'une liste	185
Chapitre 11. Données mutables	189
11.1. Introduction à la programmation impérative	189
11.1.1. Regard en arrière sur la programmation fonctionnelle	189
11.1.2. Valeurs modifiables ou mutables	190

11.2. Les références	191
11.3. Les champs mutables dans les enregistrements	194
11.3.1. Déclaration, création, consultation et modification	194
11.3.2. Retour sur les références	198
11.3.3. Synonymie, partage	199
11.4. Tableaux	200
11.4.1. Vecteurs	201
11.4.2. Analogie chaîne de caractères, vecteur	203
11.4.3. Matrices	205
11.5. Impression à l'écran	207
11.6. Séquences	209
11.7. Boucles	211
11.7.1. Boucle while	211
11.7.2. Boucle for	211
11.7.3. La factorielle : version «impérative»	213
11.7.4. Boucles et tableaux : quelques exemples	214
11.8. Résolution de systèmes linéaires par l'algorithme du pivot de Gauß	219
11.9. Recherche et tri dans un vecteur	223
11.9.1. Recherche dans un vecteur	224
11.9.2. Tris	226
11.10. Fichiers et opérations de base	232
Chapitre 12. Exceptions	239
12.1. Fonctions partielles et exceptions	239
12.2. Déclaration d'une exception	240
12.3. Levée d'une exception	242
12.4. Traitement d'une exception	244
12.5. Exception et ordre d'évaluation	247
12.6. Utilisation avancée des exceptions	248
12.6.1. Recherche d'un élément dans un tableau	249
12.6.2. Acquisition sûre de données	250
12.6.3. Un algorithme à essais successifs	251
Chapitre 13. Introduction à la modularité	253
13.1. Développement de gros logiciels	253
13.2. Modules	255
13.2.1. Définition d'un module	255
13.2.2. Impacts sur le développement de logiciels	255
13.3. Syntaxe des modules	257
13.3.1. Forme des déclarations dans l'interface	257
13.3.2. Un premier exemple de module	258
13.3.3. Modules de la bibliothèque standard	258
13.4. Utilisation des ressources d'un module	260
13.4.1. Notation pointée	260

13.4.2. La directive open	261
13.5. Compilation séparée	261
13.6. Utilisation interactive des modules	264
13.7. Types abstraits	264
Chapitre 14. Choix des structures de données	269
14.1. Structures de données linéaires	269
14.1.1. Comparaison vecteur-liste	269
14.1.2. Représentation des pires	271
14.1.3. Représentation des files	273
14.1.4. Listes chaînées	277
14.1.5. Retour sur la représentation des files	283
14.1.6. Conclusion sur les structures de données linéaires	284
14.2. Structures de données arborescentes	284
14.2.1. Arbres	284
14.2.2. Files de priorité	289
14.2.3. Tas	290
14.3. Autres structures de données	296
Deuxième partie. Quelques développements complets	297
Chapitre 15. Alignement de séquences ADN	299
15.1. Le problème posé	299
15.2. Quelques éléments de vocabulaire sur les séquences	300
15.3. Représentation des séquences d'ADN	301
15.4. Opérations d'édition	301
15.5. Alignement	302
15.5.1. Affichage d'un alignement	302
15.5.2. Application d'un alignement à une séquence	304
15.5.3. Coût d'un alignement	304
15.6. Alignement optimal et distance d'édition	305
15.6.1. Calcul de la distance d'édition	306
15.6.2. Calcul d'un alignement de coût optimal	309
15.7. Un algorithme efficace d' alignement de deux séquences	311
15.7.1. Calcul de la matrice D	312
15.7.2. Recherche d'un alignement optimal	315
15.8. Conclusion	317
Chapitre 16. Vérification de formules logiques	319
16.1. Le rôle de la logique en informatique	319
16.2. Le calcul propositionnel	319
16.2.1. Syntaxe du calcul propositionnel	320
16.2.2. Représentation des formules en <i>OCaml</i>	321
16.2.3. Tables de vérité	322
16.2.4. Interprétation d'une formule	323
16.2.5. Tautologie/contradiction	324

16.3. Analyse syntaxique de formules	328
16.3.1. Syntaxe concrète des formules	329
16.3.2. Conception descendante de l' analyseur syntaxique	330
16.4. Preuve de formules propositionnelles en déduction naturelle	340
16.4.1. Preuve formelle et règles de déduction	341
16.4.2. Tactique de preuve et vérification	343
16.4.3. Un exemple	346
16.5. Réalisation en OCaml de l' assistant à la preuve	348
16.5.1. Type des tactiques	348
16.5.2. Représentation d'un but	349
16.5.3. Affichage d'un but	349
16.5.4. Application d'une tactique à un but	353
16.5.5. Analyse syntaxique d'une tactique	357
16.5.6. Vérification interactive	360
16.6. Conclusion	363
Chapitre 17. Un exemple graphique : le jeu Court-circuit	365
17.1. Description du jeu	365
17.2. Représentation du jeu	366
17.3. Remplissage de la zone de jeu	369
17.3.1. Générateur aléatoire	369
17.3.2. Principe de remplissage de la zone de jeu	369
17.3.3. Création du tableau : appariement des couleurs	369
17.3.4. Permutation du tableau : mélange des couleurs	370
17.3.5. Affichage textuel de la zone de jeu	371
17.4. Détection d'un circuit valide et suppression d'un couple de pièces	372
17.4.1. Détection de circuits valides	372
17.4.2. Angles d'un circuit	378
17.5. Affichage graphique statique de la zone de jeu et des circuits	380
17.5.1. Présentation de la bibliothèque graphique	380
17.5.2. Dessin des pièces	383
17.5.3. Affichage graphique de la zone de jeu	388
17.5.4. Affichage graphique des circuits	389
17.5.5. Effaçage d'une pièce sélectionnée et d'un circuit	390
17.6. Affichage graphique avec gestion de l'action du joueur	391
17.6.1. Premiers pas en gestion événementielle	391
17.6.2. Interaction avec le joueur	393
17.6.3. La version de base du jeu	395
17.6.4. Version de base modulaire	397
17.7. Enrichissement du jeu avec des niveaux	401
17.7.1. Représentation du jeu	401
17.7.2. Circuits	405
17.7.3. Interaction avec le joueur	405

17.7.4. Affichage du jeu	405
17.7.5. Lancement du jeu	408
17.8. Aide au joueur	409
17.8.1. Suggestion	409
17.8.2. Backtrack	411
17.8.3. Arrêt entre les niveaux	412
17.9. Jeu contre la montre et pause	414
17.9.1. Une première version à temps compté	415
17.9.2. Version avec indication de la progression du jeu	416
17.9.3. Version avec pause	419
17.10. Scores, table des meilleurs scores	421
17.10.1. Scores	422
17.10.2. Table des meilleurs scores	424
17.10.3. Fin du jeu	431
17.11. Améliorations	433
17.12. Conclusion	438
Bibliographie	439
Index	441