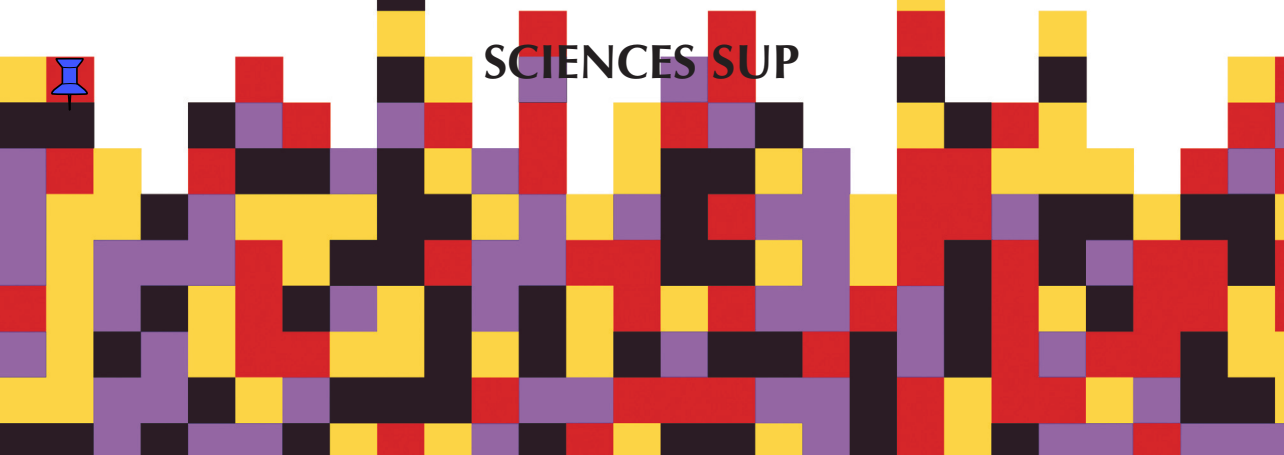




SCIENCES SUP

Cours et exercices corrigés

Licence • IUT • BTS • Écoles d'ingénieurs

PROGRAMMATION ET ALGORITHMIQUE EN VBA POUR EXCEL

*Anne Brygoo
Maryse Pelletier
Michèle Soria
et Séverine Dubuisson*

DUNOD

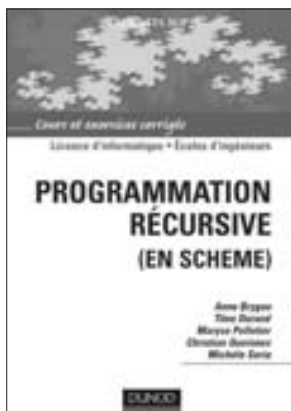
**PROGRAMMATION
ET ALGORITHMIQUE
EN VBA POUR EXCEL**

Consultez nos catalogues sur le Web



www.dunod.com

Des mêmes auteurs :



Programmation récurse (en Scheme)
 Anne Brygoo, Titou Durand,
 Maryse Pelletier, Christian Queinnec,
 Michèle Soria
 352 p.
 Dunod, 2004.

PROGRAMMATION ET ALGORITHMIQUE EN VBA POUR EXCEL

Cours et exercices corrigés

Anne Brygoo

Maître de conférences à l'université Paris 6

Maryse Pelletier

Maître de conférences à l'université Paris 6

Michèle Soria

Professeur à l'université Paris 6

et Séverine Dubuisson

Maître de conférences à l'université Paris 6

DUNOD

VBA® et Excel® sont des marques déposées de Microsoft Corporation. Toutes les autres marques citées dans cet ouvrage sont des marques déposées par leurs propriétaires respectifs.

Illustration de couverture réalisée
à partir de l'image générée
par le jeu de Same (chapitre 9)

Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.

Le Code de la propriété intellectuelle du 1^{er} juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements

d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour

les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée. Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du

droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).



© Dunod, Paris, 2007
ISBN 978-2-10-050799-3

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

Table des matières

Introduction	1
1 Noyau de Visual Basic	7
1.1 Aspects lexicaux	7
1.2 Constantes et variables	8
1.3 Fonctions en VBA	10
1.4 Structure de contrôle : séquence	14
1.5 Structure de contrôle : alternative	15
1.6 Exemple de synthèse	16
1.7 Exercices corrigés	18
2 Récursion et boucles	23
2.1 Récursion	23
2.2 Itération : boucle For	25
2.3 Itération : boucle Do	29
2.4 Choix entre différents types de boucle	33
2.5 Exercices corrigés	34
3 Procédures et entrées/sorties	39
3.1 Procédures	39
3.2 Macros	41
3.3 Entrées/sorties	43
3.4 Exercices corrigés	49
4 Objets en VBA Excel	57
4.1 Notion d'objet	57
4.2 Notion de classe	58
4.3 Notions de propriété et méthode	59
4.4 Notation pointée	59
4.5 Spécification et utilisation des propriétés et des méthodes	60
4.6 Instruction With	62
4.7 Notion de collection	63
4.8 Propriétés, fonctions et procédures globales	64
4.9 Classe Worksheet	65
4.10 Classe Range	66
4.11 Déclaration et affectation d'objets	70
4.12 Formatage d'une plage de cellules	71
4.13 Macros enregistrées	75
4.14 Exercices corrigés	76

5	Programmation de calculs	89
5.1	Traitements sous le tableur	90
5.2	Traitements par programmes	93
5.3	Conclusion	102
5.4	Exercices corrigés	102
6	Algorithmes de recherche	109
6.1	Présentation du problème	109
6.2	Recherche séquentielle	111
6.3	Recherche dichotomique	117
6.4	Exercices corrigés	121
7	Algorithmes de tri	131
7.1	Présentation du problème	131
7.2	Tri par insertion	132
7.3	Tri par sélection	137
7.4	Tri rapide (Quicksort)	142
7.5	Exercices corrigés	147
8	Algorithmes de mise à jour	155
8.1	Recalcul des valeurs	155
8.2	Recalcul des cellules	162
8.3	Exercices corrigés	166
9	Problèmes de synthèse	173
9.1	Énoncés	173
9.2	Corrigés	183
Annexes		199
A	Éléments sur le tableur Excel	199
A.1	Introduction	199
A.2	Formules	200
A.3	Références	202
A.4	Erreurs	204
A.5	Exercices corrigés	206
B	Fiche technique Excel	214
B.1	Présentation de l'environnement Excel	214
B.2	Travailler dans la feuille de calcul	215
C	Fiche technique VBA	217
C.1	Présentation de l'environnement VBE	217
C.2	Écriture de programmes en VBA	218
C.3	Pour obtenir de l'aide	220
C.4	Une première exécution étape par étape	220
D	Mémento de Visual Basic	221
D.1	Syntaxe de Visual Basic	221
D.2	Domaine de programmation	224

Introduction

Depuis les débuts de l'enseignement de la discipline informatique, il y a une cinquantaine d'années, les cours dits d'introduction font l'objet de débats intenses, tant sur leur structure que sur leur contenu. De nombreuses expériences ont été développées, mais il n'y a pas encore de consensus sur une stratégie optimale : est-il préférable de commencer par les concepts théoriques ou par l'apprentissage d'un langage de programmation, et dans ce cas, quel langage choisir ? Chaque méthode présente des avantages et des inconvénients, et le choix dépend aussi du but recherché, de l'auditoire, des ressources...

Contexte

Notre approche est orientée langages. Nous intégrons la théorie et la pratique et nous insistons sur l'importance tant de l'abstraction que de l'implantation : nous dégagons des principes de programmation et nous montrons comment les programmer de manière effective. Il nous paraît indispensable d'apprendre très vite plusieurs langages de programmation appartenant à des familles différentes (fonctionnel, impératif...) : c'est la diversité des expériences qui permet de percevoir ce qui est invariant et ce qui change, de faire la part entre les principes fondamentaux, les différents modes de pensée, les constructions générales et les détails syntaxiques.

L'environnement de programmation aussi nécessite un choix ; il nous paraît tout aussi important de savoir construire ses programmes brique à brique, dans un environnement initial qui ne contient rien, que de travailler dans un environnement déjà très riche sans se perdre. Programmer, c'est non seulement construire ses propres programmes, mais aussi savoir utiliser, voire modifier, ce qui a été fait par d'autres.

Ce livre est le troisième d'une série d'ouvrages d'introduction à la programmation (le premier traite de la représentation et de la manipulation de l'information¹ et le deuxième s'intéresse à la programmation fonctionnelle et à ses mécanismes d'évaluation²), qui décline les principes évoqués plus haut : variété des modèles de programmation et des langages, variété des environnements, mais cohérence des principes. L'entreprise a démarré il y a une dizaine d'années, à l'occasion d'une réforme du premier cycle à l'Université Pierre et Marie Curie (UPMC), et ces ouvrages servent de base dans le cursus de la Licence d'informatique.

Ce livre sur la programmation impérative correspond à un module du second semestre de Licence à l'UPMC, dans les parcours « Mathématique, Informatique, Mécanique, Electronique » ; mais de par son contenu, il est aussi utilisé dans des formations de plus haut niveau (Ecole d'ingénieur ou Master), pour donner à des

¹Initiation à la programmation par Word et Excel, *M. Soria, A. Brygoo, M. Morcrette, O. Paliès, Vuibert.*

²Programmation récursive (en Scheme), *A. Brygoo, T. Durand, M. Pelletier, C. Queinnec, M. Soria, Dunod.*

étudiants non spécialistes en informatique des bases solides en programmation, tout en leur permettant de valoriser leur savoir-faire sur le marché du travail.

Objectifs

L'objectif premier de ce livre est de présenter les traits principaux de la programmation impérative. Nous avons choisi de travailler avec VBA pour Excel³, car il est possible d'écrire très vite des programmes intéressants (cependant le lecteur doit avoir une connaissance minimale des fonctionnalités d'un tableur, et nous avons ajouté une synthèse des éléments nécessaires en annexe A).

Ce livre n'est pas une présentation exhaustive du langage VBA. Il a pour originalité de proposer une vision concise et épurée de la programmation impérative, en utilisant VBA. Pour illustrer les principes et les concepts que nous voulions mettre en avant, nous avons extrait du domaine de programmation de VBA, d'une part les instructions minimales pour enseigner les mécanismes impératifs, et d'autre part une cinquantaine de méthodes, suffisantes pour présenter les principaux traitements et travailler sur des applications motivantes.

L'utilisation d'un environnement de programmation dédié permet de bénéficier de l'existant : en VBA pour Excel, les objets du tableur ont une représentation effective en mémoire et il existe un ensemble de primitives (propriétés et méthodes) pour manipuler ces objets. Dans cet environnement, le débutant peut, très rapidement, écrire des programmes intéressants, réalistes ou ludiques : la feuille de calcul permet de traiter de nombreux problèmes modélisables à l'aide d'un tableur, et elle peut aussi servir de table graphique pour programmer des jeux.

Ce livre est aussi une initiation à l'algorithmique : nous étudions tout d'abord le problème de la recherche d'un élément dans un ensemble, puis celui du tri d'un ensemble d'éléments, en comparant les performances de différents algorithmes de résolution. Pour terminer, nous revenons sur le tableur pour une meilleure compréhension des mécanismes sous-jacents (et donc une utilisation plus experte) : nous traitons le problème de la mise à jour automatique des valeurs de la feuille de calcul, en présentant la structuration des données et les algorithmes permettant un recalcul efficace lors de la modification d'une cellule.

Notre choix s'est porté sur Excel et VBA car ce sont des logiciels très répandus. Mais les concepts que nous introduisons s'appliquent à bien d'autres tableurs et langages de macros, et l'étudiant qui aura assimilé ce livre pourra très vite s'adapter à un autre contexte de tableur. De plus, la démarche que nous avons suivie peut s'appliquer à d'autres types d'applications logicielles (traitement de texte, base de données...); non seulement nous pensons qu'il est intéressant de construire des enseignements du même type autour d'autres applications, mais nous croyons aussi que l'étudiant qui aura intégré l'esprit de ce livre, aura des repères précis pour aborder la programmation de macros dans des contextes logiciels très différents.

Organisation de ce livre

Chapitres – Ce livre s'organise en neuf chapitres, regroupés en quatre volets.

Les cinq premiers chapitres sont consacrés à l'étude de la programmation. Le chapitre 1 est une présentation des bases de la programmation en Visual Basic, qui permet d'écrire les premières fonctions, en utilisant l'affectation, la séquence et

³VBA (Visual Basic for Applications) et Excel sont des marques déposées Microsoft.

l'alternative. Le chapitre 2 est consacré à l'étude de l'itération et présente de façon détaillée les différentes sortes de boucles et leurs implantations. Le chapitre 3 introduit les procédures et les macros, ainsi que les interactions entre le programme et l'utilisateur. Le chapitre 4, qui est une introduction à l'utilisation des objets, montre comment utiliser les classes de VBA sur l'application tableur. Le chapitre 5 termine cette première partie en illustrant toutes les notions présentées antérieurement, sur une étude de manipulation de matrices.

Le deuxième volet aborde le domaine de l'étude des algorithmes. En travaillant toujours sur la feuille de calcul du tableur, on présente le problème de la recherche d'un élément dans une séquence (chapitre 6) et celui du tri d'une séquence d'éléments (chapitre 7). On étudie différentes méthodes, différents algorithmes et différentes implantations, en les comparant du point de vue de leur efficacité.

Le troisième volet (chapitre 8), est consacré au problème de la cohérence des informations affichées sur une feuille de calcul : lorsque le contenu d'une cellule est modifié, le tableur doit recalculer toutes les cellules impactées par cette modification. L'étude de l'algorithme de mise à jour permet de mettre en évidence le graphe de dépendances qui structure la feuille de calcul ; nous introduisons, à cette occasion, quelques notions fondamentales sur les graphes.

Enfin le dernier volet (chapitre 9), présente un éventail de problèmes, appartenant à des domaines divers, qui font mettre en pratique toutes les notions importantes présentées dans le livre.

Exercices – On apprend à programmer... en programmant. Nous essayons ici de faire comprendre les principes par l'exemple.

Chaque partie de « cours » est émaillée d'exemples qui permettent d'illustrer les nouvelles notions et de mettre en avant différentes méthodes et styles de programmation. Pour s'appropriier ces notions, il est important d'implanter les solutions proposées, en utilisant l'interaction avec la machine pour reproduire ce qui a été fait, expérimenter et tester des variations, résoudre des problèmes voisins...

Chaque chapitre propose aussi un bon nombre d'exercices corrigés, rangés par ordre de difficulté croissante. Il faut essayer de les traiter sans regarder la solution (les corrigés sont reportés en fin de chapitre à cette intention), implanter sa propre méthode et la tester. Le bénéfice que l'étudiant tirera des explications et des programmes donnés en corrigé sera d'autant plus grand qu'il aura vraiment cherché à travailler l'exercice par lui-même, qu'il se sera confronté à des difficultés et aura tenté de les résoudre à sa manière.

Enfin un chapitre entier est consacré à la résolution de problèmes mettant en œuvre tout ce qui a été présenté auparavant (sans isoler chaque notion, comme cela est fait dans les exercices). On apprend alors à modéliser un problème et à rechercher et mobiliser les connaissances acquises.

Nous accordons aussi une grande importance à la présentation des programmes, dans un but de clarté et de lisibilité. La spécification, le choix des noms, la structuration, obéissent à des règles très précises, qui sont mises en évidence par de nombreux enrichissements typographiques (gras, italique, indentation...).

Annexes – Différents documents sont proposés en annexes de ce livre : une présentation des éléments de base d'un tableur, des fiches techniques pour les environnements de travail, et un manuel de référence pour l'écriture des programmes.

Une connaissance minimale des fonctionnalités d'un tableur et de la manipulation des formules est un prérequis nécessaire pour pouvoir écrire des programmes

VBA sous Excel. Une présentation succincte des éléments de base (écriture des références, des formules...) est donc donnée en annexe.

Les fiches techniques donnent des informations pratiques sur les environnements de travail. La fiche technique Excel explique comment manipuler les classeurs, feuilles de calcul et cellules. La fiche technique VBA explique comment utiliser l'environnement de programmation pour écrire, exécuter et déboguer des programmes.

L'annexe *mémento*, donne la syntaxe de Visual Basic ainsi que la liste des propriétés et méthodes de VBA utilisées. Le mémento regroupe toutes les connaissances de programmation exposées dans ce livre. Ce mémento n'est pas un résumé de VBA pour Excel (même s'il peut être utilisé pour une introduction à VBA). Pour exprimer les programmes, nous nous limitons à un sous-ensemble restreint et structuré du langage VBA, tant au niveau des constructions syntaxiques qu'au niveau des propriétés et méthodes. Depuis sa première version, le mémento a fait l'objet de dizaines de restructurations et de centaines de modifications plus ou moins importantes. Pourquoi tant de mises au point ont elles été nécessaires? Bien sûr pour corriger des erreurs mais surtout pour assurer la cohérence du document, reflet de notre enseignement, et éviter de se disperser dans le foisonnement du catalogue de méthodes et propriétés de VBA. Le mémento est le fil conducteur pour tous les participants à l'enseignement, tant les enseignants – une quarantaine chaque année, que les étudiants – 700 à 800 chaque année. Les enseignants s'en servent comme référence pour les principes et les connaissances à transmettre; et les étudiants y trouvent toutes les informations nécessaires pour traiter les devoirs et les examens.

Site compagnon – Le site compagnon de ce livre est indiqué sur la page suivante : www.dunod.com. On pourra y télécharger différents autres documents : correctifs, compléments sur les logiciels, codes sources et exercices supplémentaires.

Remerciements

En premier lieu nos remerciements profonds vont à Titou Durand, sans qui ce livre n'existerait probablement pas. Titou a été pendant plusieurs années l'un des piliers de la construction des cours de programmation à l'UPMC, tant au niveau de la réflexion que de la mise en œuvre. Son expérience et ses compétences sont à la base de bon nombre des choix pédagogiques que nous avons faits (en général après des heures de discussions aussi animées que chaleureuses!), et de nombreuses pages de ce livre sont issues de ses idées et de ses réalisations. Son départ à la retraite ne lui a pas donné la possibilité de participer à la phase de rédaction proprement dite, mais il a toujours été au bout du fil pour nous aider à résoudre un problème technique ou pédagogique. La version définitive du livre a bénéficié de sa relecture exigeante.

Nous remercions aussi tous nos collègues de l'UPMC, avec lesquels nous avons enseigné depuis une dizaine d'années. Leur collaboration enthousiaste a permis de monter un enseignement original, affiné au fil des ans par leurs questions et leurs suggestions; et de dispenser cet enseignement à une dizaine de milliers d'étudiants... ce qui est parfois un tour de force.

Nos remerciements s'adressent également à la hiérarchie de l'UPMC, qui a toujours soutenu nos expériences pédagogiques et nous a permis de réaliser différents développements logiciels autour d'Excel.

Nous tenons enfin à remercier Christian Queinnec et les éditions Paracamplus qui ont édité une version préliminaire de cet ouvrage au printemps 2006.

Avertissement au lecteur

Dans le cursus de l'UPMC, l'enseignement de programmation impérative en VBA vient à la suite d'un cours de programmation fonctionnelle en Scheme, et le présent ouvrage fait suite à celui sur la programmation récursive évoqué plus haut. Nous essayons de maintenir une forte homogénéité dans la présentation des traits de programmation et dans l'écriture des programmes, en particulier au niveau des spécifications. Pour passer de la programmation fonctionnelle à la programmation impérative, il faut cependant comprendre quelques différences essentielles existant entre ces deux approches, au niveau de l'environnement d'exécution, des entités élémentaires (expressions et instructions), des mécanismes de répétition (récursion et itération), des deux types de sous-programmes (procédures et fonctions). Enfin l'utilisation de VBA permet d'approcher la programmation orientée objet, en manipulant les classes qui existent pour l'application Excel.

Nous présentons ici une synthèse des principaux concepts de programmation abordés ; l'étudiant débutant peut passer cette section en première lecture, mais il est invité à y revenir régulièrement lors de son étude des quatre premiers chapitres.

Spécification des programmes – La *spécification* d'un programme définit les termes du contrat qui lie l'utilisateur et le réalisateur du programme. La spécification d'un programme contient deux parties :

- la *signature* (nom, liste et type des arguments, type du résultat pour une fonction), définit la syntaxe d'emploi ; elle est donnée dans l'en-tête du programme ;
- la *sémantique* décrit ce que fait le programme (et non comment il le fait) ; elle est exprimée sous forme de commentaires, mais on s'attachera à respecter des règles systématiques : expliciter les hypothèses et les erreurs dans des clauses particulières, expliciter le résultat de l'exécution du programme (renvoi d'une valeur pour une fonction et modification de l'environnement d'exécution pour une procédure).

Environnement – Lorsque l'on programme dans un langage purement fonctionnel, l'*environnement* est constitué de définitions de fonctions et de valeurs de variables, ces dernières ayant une valeur constante tant qu'elles sont présentes dans l'environnement.

En programmation impérative, cette notion d'environnement est rendue encore plus importante, par la possibilité de changer la valeur d'une variable pendant l'exécution du programme. Ainsi, alors qu'en programmation purement fonctionnelle, on peut seulement *enrichir* l'environnement par de nouvelles définitions, en programmation impérative on peut également *modifier* l'état de l'environnement par la modification des valeurs des variables au cours de l'exécution du programme – on parle alors d'environnement d'exécution.

Expressions et instructions – En style purement fonctionnel, on ne fait que composer des expressions et tout programme est décrit par une expression. En style impératif, un programme est un enchaînement d'instructions à exécuter. On manipule alors deux types d'entités élémentaires, les expressions et les instructions :

- une *expression* s'évalue en un résultat, d'un certain type ; par exemple $3+5-7$ ou `factorielle(5)` sont des expressions entières et $X < 0$ est une expression booléenne ;
- une *instruction* décrit une action de l'ordinateur ; par exemple l'appel de procédure `effacerTout` et l'affectation $X = X + 1$ sont des instructions.