

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