

 **L'intro** 

Programmation

Microsoft

Access[®] 2000

Paul Kimmel

- ▶ **Maîtrisez la programmation VBA pour Access 2000**
- ▶ **Apportez vos propres modifications sur tous les éléments du logiciel**
- ▶ **Apprenez à intégrer simplement des bases de données à vos pages Web**



CAMPUSPRESS

2-005-442-1

2-005-442-1

L'intro

Table des matières

Programmation Access 2000

Paul Kimmel



CAMPUSPRESS

Table des matières

Introduction	3
--------------------	---

Partie 1. INTRODUCTION À LA PROGRAMMATION POUR ACCESS

1. Nouveautés d'Access 2000	7
Personnalisation des menus et des barres d'outils	8
La commande de menu Plus	8
Personnaliser les menus et les barres d'outils	9
Associer un lien hypertexte à une commande	10
Vérification de l'orthographe des enregistrements	11
Collaboration en ligne	13
Configuration de NetMeeting	13
Une fois connecté... ..	14
Introduction aux modèles d'objets ADO (ActiveX Data Objects)	15
Le modèle ADOX, champion de la sécurité	15
Les procédures stockées dans Access 2000	15
La programmation orientée objet pour Access 2000	16
Ajout de données à une page Web	16
Les projets Access	17
2. Éléments de base du langage VBA	19
Fonctionnement de VBA pour Access	20
Mots clés de VBA pour Access	21
Opérateurs et opérandes d'Access	23
Données connues d'un programme	25
Révisions et passage à la pratique	28
Opérateurs arithmétiques	28
Opérateurs de comparaison	29
Opérateurs logiques	30
Opérateurs de concaténation de chaîne	31
Prélude à des sujets plus sérieux	32



3. Stockage des données d'un programme : les variables	35
Fonctionnement de la mémoire dans un ordinateur	36
Déclarer des variables	37
Variables Dim	38
Variables ReDim	38
Variables Const	39
Variables globales	40
Affectation des variables et évaluation de leur valeur	41
De l'intérêt d'être explicite... dans la déclaration des variables	42
Le type de données générique Variant	43
La déclaration implicite des variables	43
Garde-fou : l'instruction Option Explicit	43
Les info-bulles de VBA	44
Elaboration d'un programme d'essai	45
Déboguer un programme et utiliser les info-bulles	46
La fenêtre Variables locales	47
La fenêtre Espions	48
Utilisation de la fenêtre Espions	49
Modification des espions	50
L'outil Espion express	51
Exécuter du code dans la fenêtre Exécution	51
4. Les opérateurs en programmation : évaluations et calculs	55
Eléments d'une équation	56
Données d'une équation	56
Règles de conduite	59
Nombre d'opérandes des opérateurs	60
Utilisation des opérateurs arithmétiques	61
Les opérations de comparaison	62
Tables de vérité et opérateurs logiques	63
Les opérations bit par bit	68
Concaténation de chaîne	70
Règles de priorité des opérateurs	71
Opérateurs particuliers	72
Is : comparer deux variables objet	73
Like : tester le motif d'une chaîne	73
AddressOf : renvoyer l'adresse d'une procédure	73



Partie 2. GESTION DES DONNÉES ACCESS PAR LA PROGRAMMATION

5. Instructions conditionnelles et boucles	77
Au garde-à-vous !	78
L'instruction Else	80
Imbrication de constructions conditionnelles	81
Ordre et rigueur	82
Alternatives aux constructions If imbriquées : l'instruction Select Case	83
Première construction de boucle : While...Wend	85
Les constructions de boucle Do...Loop	86
Traiter itérativement des données : la boucle For...Next	89
Spécifier un pas d'itération	90
Les fonctions Ubound et Lbound	92
Traiter les éléments d'une collection avec la boucle For Each...Next	93
Sortir d'une boucle avec l'instruction Exit	94
6. Gérer une base de données	97
Création d'une base de données	98
Définition de la base de données	98
Création d'une table dans Access 2000	98
Création d'un module dans la base de données Contacts	101
Création d'une table par du code	102
Gestion d'une table avec du code	105
Quelques rappels terminologiques sur les bases de données	106
Accéder aux données d'une table	106
Remplissage d'une table enregistrement par enregistrement	109
Recherche conditionnelle de données dans une table	111
7. Types de données avancés : objets OLE, ADO et DAO	113
Qu'est-ce que l'automation OLE ?	114
L'univers des objets	115
Objets utiles	116
Comparaison des objets ADO et DAO	117
Utilisation de l'objet ADODB	118
Utilisation d'une connexion (Connection)	119
Utilisation de l'objet Recordset	121
Le modèle d'objet ADOX	125
L'objet Catalog	125



8. Sous-routines et fonctions : décomposer une tâche en sous-tâches ..	131
Technique d'écriture des sous-routines	132
Instructions de début et de fin d'une sous-routine	133
Conventions de nommage	134
Nommage des arguments d'une sous-routine	134
Déclaration des arguments et de leurs types de données	135
Technique d'écriture des fonctions	138
Principes à respecter en programmation	140
Règle n°1 : Mettre les blocs de code répétés sous forme de fonctions séparées ..	140
Règle n°2 : Veiller à la concision de chaque fonction	140
Règle n°3 : Limiter le nombre d'arguments	141
Règle n°4 : Recourir aux qualificatifs d'argument comme garde-fous contre les modifications intempestives de variables	141
Règle n°5 : Préciser à l'utilisateur les conditions à réunir pour le bon fonctionnement du code	141
Règle n°6 : Commenter son code abondamment	142
Application pratique	143
Création d'une table avec du code	143
Importation d'un fichier de valeurs séparées par des virgules (.CSV)	144
Recherche d'un enregistrement	146
Utilisation de Windows API pour lire dans la base de registres	147

Partie 3. EXPLORATION DES OUTILS DE PROGRAMMATION EXISTANTS

9. Les macros, outils pédagogiques par excellence	151
B.A.-BA de la création de macro	152
Création d'une table avec du langage SQL	152
Création d'une macro	153
Nom des macros et conditions d'exécution	155
Test et débogage d'une macro	157
L'action CopierObjet	159
L'action SupprimerObjet	160
Marge de manœuvre étendue grâce à l'action ExécuterCommande	162
Importer des données	163
Transférer une base de données : l'action TransférerBase	164
Transférer des données texte : l'action TransférerTexte	165
Utilisation des actions de macro dans du code	169
Macros : pour ou contre	170



10. Les solutions prêtes à l'emploi d'Access	171
La manipulation de chaînes	172
Conversion entre chaînes et valeurs numériques	172
Les fonctions de chaîne ambivalentes	174
Comparaison de chaînes	177
Allocation dynamique de chaînes	179
Appliquer des formats à des chaînes	179
Les fonctions Date, Time et Now	181
Lecture/écriture dans des fichiers	181
Ouverture et fermeture de fichiers avec les fonctions Open et Close	182
Lecture et écriture de données texte	186
Lecture et écriture de données binaires	189
Les fonctions d'interaction dynamique avec l'utilisateur	190
Rechercher des fonctions, méthodes et propriétés dans VBA	191

Partie 4. TYPES DE DONNÉES PERSONNALISÉS, TABLEAUX ET COLLECTIONS

11. Création de types de données personnalisés : vers la simplification du code	195
Digression philosophique : élaboration de concepts	196
Déclaration d'un type de données nouveau avec l'instruction Type	197
Types de données pouvant être utilisés pour créer un type défini par l'utilisateur ...	199
Déclaration des instances de type utilisateur	201
Création d'un type à constantes prédéfinies avec la déclaration Enum	202
Pour résumer	205
12. Les tableaux en programmation	207
Qu'est-ce qu'un tableau ?	207
Déclaration d'un tableau	209
Déclarer des tableaux de dimension fixe	209
Déclarer des tableaux dynamiques	210
Déclarer des tableaux statiques	212
Premier indice d'un tableau : 0 ou 1 ?	212
Application des tableaux au tri de données	213
Fonctions de manipulation de tableaux	216
Création d'une fonction de création de tableau	218



Règles à respecter lors de l'utilisation de tableaux	219
Tri des données d'un tableau	219
Algorithme de tri à bulles	220
Algorithme de tri à bulles sélectif	221
Algorithme de tri rapide	222
13. Les collections	227
Comprendre les collections	228
Applications des collections	228
Terminologie relative à la programmation orientée objet	229
Déclaration d'objets Collection	230
Gestion d'une collection de données	233
Ajouter des éléments à une collection	233
Soustraire une donnée d'une collection	236
La méthode Item et la propriété Count	236
Comment reconnaître une collection	238

Partie 5. CRÉATION D'APPLICATION DE BASES DE DONNÉES ACCESS

14. Organisation de la complexité en programmation	241
Conventions de nommage en programmation	242
Le sens passe par les mots	244
La structure verbe-complément pour nommer les sous-routines et les fonctions	245
Éviter les abréviations fantaisistes	245
Mise en forme du code avec des espaces	246
Organisation de la complexité	248
Éviter les structures imbriquées	248
Mettre les instructions conditionnelles dans des procédures séparées	249
Brièveté et pertinence des procédures	250
L'art de commenter un code	251
Commenter avec des phrases entières	251
Commenter les codes longs ou importés d'autres sources	251
Avantages de la réutilisation de code	251
Test et débogage	252
Règles de manipulation des données d'un programme	253



15. Les objets ADODB	255
Connexion à une base de données	256
Les paramètres d'ouverture de connexion	258
Création d'un alias ODBC	260
Qu'est-ce qu'un fournisseur ?	262
Spécifier un nom d'utilisateur et un mot de passe	263
Sélectionner un mode de connexion	263
Gestion de l'objet Recordset	264
Ouvrir et fermer un objet Recordset	265
Ingrédients de l'objet Recordset	267
Ajouter et modifier des données	269
Modifier les données d'un enregistrement	272
Supprimer des données	272
Délimitation des enregistrements cibles	274
Utilisation d'une table	274
Utilisation d'une requête	276
Utilisation de la propriété Filter	278
Ajouter des données à une collection	278
Informations en ligne sur le modèle ADO (ActiveX Data Objects)	280
16. Techniques SQL avancées	281
L'instruction Select	282
Forme simple de l'instruction SELECT	283
La clause WHERE pour filtrer l'objet Recordset	284
Opérateurs possibles de la clause WHERE	286
Instructions SELECT imbriquées dans des clauses WHERE	288
Requêtes de tri	289
Regrouper par champ	289
Filtrer les résultats avec la clause HAVING	290
Les jointures de tables	291
Création d'une UNION de tables	294
Définir un pseudonyme (alias) de colonne	295
Insertion de données nouvelles	295
Insérer des données dans des champs spécifiés	296
Insérer des données dans tous les champs par défaut	296
Insérer des données dynamiquement	297
Importer des données à partir d'une autre table avec la commande SELECT	298



Modification des données	299
Suppression de données	300
Appel de fonctions à partir de code SQL	301
Les procédures stockées	302
Ajouter une procédure stockée à un catalogue	304
Exécuter une procédure stockée	305

Partie 6. GESTION D'ERREURS ET DÉBOGAGE

17. Le débogage	309
Ecrire des codes de test	310
Des codes de test à la création du code	311
Objet du test	311
Echafauder un code de test dans Access	312
Marquer l'exécution d'instructions	314
Qu'est-ce que du code témoin ?	315
Utilisation de trappes	315
Espionner l'exécution du programme	316
Vérifier les conditions d'exécution : les points d'arrêt	318
La programmation "contractuelle"	319
Réutilisation des codes de débogage	320
Les directives de compilation	320
Mesures de sécurité quant au code auxiliaire	324
Deux objectifs : diagnostic et amélioration de la robustesse	325
18. La gestion d'erreurs	327
Comparaison des différents styles de gestion d'erreurs	328
Prévention d'erreurs conditionnelle : à l'ancienne	328
Gestion d'exceptions : une technique plus éprouvée	329
Création d'un gestionnaire d'erreurs	330
Instruction d'ouverture d'un gestionnaire d'erreurs	330
Quelques conseils sur la création de gestionnaires d'erreurs	332
Désactivation d'un gestionnaire d'erreurs	332
La gestion d'erreurs silencieuse : ignorer les erreurs	333
Recommencer avec l'instruction Resume	333
Poursuivre avec l'exécution avec Resume Next	335



L'objet Err	336
Propriétés de l'objet Err	336
Méthodes de l'objet Err	336
Validation des données utilisateur	337
Préservation des ressources mémoire	339
Rappels sur l'objet Debug	340

Partie 7. CRÉATION D'INTERFACES UTILISATEUR ACCESS

19. La création de formulaires personnalisés	343
Les Assistants Formulaires	344
Les types de formulaires prédéfinis	344
Création d'un formulaire avec l'Assistant Formulaire	347
Utilisation d'un formulaire	350
Filtrage des enregistrements de formulaire	351
Personnalisation des formulaires créés avec l'Assistant	354
Les composants visuels	355
Ajouter des contrôles au formulaire	360
Ajouter du code au formulaire	362
Test d'un formulaire	363
Démarrage de votre application	363

Partie 8. LA PROGRAMMATION ORIENTÉE OBJET DANS ACCESS

20. Notion de classe en programmation orientée objet	367
De l'intérêt des classes	368
Création d'une classe	370
Accès sélectif au code : pour une utilisation plus facile du code	372
Définition des méthodes d'une classe	374
Définition des propriétés d'une classe	376
Propriétés simples et propriétés objets	379
Propriétés statiques	379
Codes d'initialisation et de fermeture d'instances	380
Exemple pratique de classe	381
Tester une classe	385
Étendre les capacités de classes existantes	385
Création d'une instance de classe	385



21. Techniques avancées de programmation orientée objet	387
Identifier l'objectif	388
Quand écrire une classe ?	388
Principes de base de la conception en programmation orientée objet	389
Identifier ce qu'une classe doit faire et les données qu'elle doit contenir	390
Astuces d'implémentation de classes	392
Incorporation de classes existantes	393
Implémentation du stockage des données avec l'objet Collection	393
Implémentation des tâches de lecture et d'écriture avec la classe GererFichier	396
Test des classes nouvelles	399
Utilisation de la classe créée	399
22. Les compléments Access	401
Introduction aux compléments	402
Création d'une base de données servant de journal d'erreurs	403
Créer la table dynamiquement par du code	404
Répertorier les erreurs dans la base de données	405
Création d'une interface d'affichage du journal d'erreurs	408
Test du complément dans VBE	408
Installation et désinstallation de compléments	410
Création de la table des informations de registre	412
Installation d'un complément	413
Désinstallation et modification d'un complément	414
Test final	415
Créer un complément : résumé de la procédure	415

