

Les éléments à télécharger sont disponibles à l'adresse suivante :

**<http://www.editions-eni.fr>**

Saisissez la référence ENI de l'ouvrage **RI12CSHAVSC** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

## Avant-propos

### Chapitre 1

#### Introduction

- 1. Qu'est-ce que C# ? ..... 9
  - 1.1 Que peut-on réaliser avec C# ? ..... 10
  - 1.2 Le langage est-il stable et pérenne ? ..... 12
- 2. Préparer son environnement ..... 13
  - 2.1 Installation et configuration de Visual Studio Code ..... 14
  - 2.2 Installer les outils de compilation ..... 17
- 3. Comment fonctionne le C# ? ..... 18

### Chapitre 2

#### Premier programme

- 1. Créer sa première application C# ..... 23
- 2. Comprendre et écrire du code C# ..... 27
  - 2.1 Notions de variable et constante ..... 29
    - 2.1.1 Types numériques ..... 31
    - 2.1.2 Types textuels ..... 34
    - 2.1.3 Valeur booléenne ..... 38
    - 2.1.4 Opérateurs ..... 38
  - 2.2 Les autres types ..... 41
    - 2.2.1 Stockage des dates ..... 41
    - 2.2.2 Les intervalles de temps ..... 43

# 2 ————— C# 12 et Visual Studio Code

Les fondamentaux du langage

3. Analyser la structure d'un projet C#	44
3.1 La notion de blocs	45
3.2 Signification des blocs de code	48
3.2.1 Le bloc d'espace de noms	48
3.2.2 La définition d'une classe	51
3.2.3 La définition d'une méthode	52
3.3 Déclaration « top-level »	52
4. Exécuter un programme C#	53
4.1 Lancer le programme avec Visual Studio Code	53
4.2 Lancer depuis la ligne de commande	55
5. Exercice	58
5.1 Énoncé	58
5.2 Corrigé	58

## Chapitre 3

### Programmation orientée objet

1. Principes de la programmation orientée objet	61
1.1 Qu'est-ce qu'une classe ?	61
1.1.1 Les classes dans Visual Studio Code	63
1.1.2 L'héritage	64
1.1.3 L'encapsulation	65
1.2 Que peut-on déclarer dans une classe ?	66
1.2.1 Les méthodes	66
1.2.2 Déclarer une donnée	69
1.3 Instancier une classe	74
1.3.1 Le constructeur	74
1.3.2 L'instanciation avec le mot-clé new	78
1.4 Le polymorphisme	80

2. Concepts avancés . . . . .	82
2.1 L'héritage avancé. . . . .	82
2.1.1 Méthodes virtuelles . . . . .	82
2.1.2 Classe abstraite . . . . .	83
2.1.3 Interface . . . . .	85
2.1.4 Implémentation par défaut dans une interface . . . . .	87
2.1.5 Masquage . . . . .	88
2.1.6 Interdire l'héritage . . . . .	89
2.2 Les différents types d'objets. . . . .	90
2.2.1 Les types références . . . . .	90
2.2.2 Les types valeurs . . . . .	91
2.2.3 Les types nullables . . . . .	94
2.2.4 Les types références nullables . . . . .	96
2.2.5 Les énumérations . . . . .	97
2.2.6 Les records . . . . .	100
2.3 Les modificateurs de classe. . . . .	103
2.3.1 La notion de static . . . . .	103
2.3.2 Static et interfaces . . . . .	105
2.3.3 La notion de classe partielle . . . . .	106
3. Exercice . . . . .	107
3.1 Énoncé . . . . .	107
3.2 Corrigé . . . . .	108

## Chapitre 4 Algorithmique

1. Bases de l'algorithmique . . . . .	111
1.1 La logique conditionnelle . . . . .	111
1.1.1 Test simple : le if/else . . . . .	112
1.1.2 Multiples tests avec l'instruction switch . . . . .	118
1.1.3 Pattern matching . . . . .	120
1.1.4 Exercice - énoncé . . . . .	125
1.1.5 Exercice - corrigé . . . . .	125

# 4 \_\_\_\_\_ C# 12 et Visual Studio Code

Les fondamentaux du langage

1.2	Les collections . . . . .	126
1.2.1	L'interface IEnumerable . . . . .	126
1.2.2	Les tableaux . . . . .	127
1.2.3	La liste . . . . .	129
1.2.4	Les dictionnaires . . . . .	132
1.2.5	Les collections algorithmiques . . . . .	134
1.3	Les boucles . . . . .	136
1.3.1	Les généralités sur les boucles . . . . .	136
1.3.2	La boucle for . . . . .	137
1.3.3	La boucle while . . . . .	139
1.3.4	La boucle do while . . . . .	139
1.3.5	La boucle foreach . . . . .	140
1.3.6	Le mot-clé yield . . . . .	140
1.3.7	Exercice - énoncé . . . . .	141
1.3.8	Exercice - corrigé . . . . .	142
2.	Gestion des erreurs . . . . .	144
2.1	Concept d'une exception . . . . .	145
2.2	Renvoyer une exception . . . . .	146
2.3	Gérer une exception . . . . .	148
2.3.1	Blocs try, catch et finally . . . . .	149
2.3.2	Filtre sur bloc catch . . . . .	151
2.4	Exceptions et performances . . . . .	153

## Chapitre 5

### LINQ

1.	Fonctionnement de base . . . . .	155
2.	Variables anonymes . . . . .	158
3.	Principes des opérateurs LINQ . . . . .	158
3.1	Opérateurs de production . . . . .	162
3.2	Opérateurs de sélection . . . . .	174
3.3	Opérateurs de génération . . . . .	180

4. Expression de requête LINQ .....	181
4.1 Le mot-clé into .....	182
4.2 Le mot-clé let .....	184
5. Exercice .....	185
5.1 Énoncé.....	185
5.2 Corrigé.....	186

## Chapitre 6 Sérialisation

1. Sérialisation en C# .....	187
2. Sérialisation XML .....	188
2.1 XmlSerializer .....	189
2.2 XmlDocument, XElement et XAttribute .....	193
3. Sérialisation JSON .....	197
3.1 Utf8JsonReader et Utf8JsonWriter .....	198
3.2 JsonDocument .....	201
3.3 JsonSerializer.....	202
4. Exercice .....	208
4.1 Énoncé.....	208
4.2 Corrigé.....	209

## Chapitre 7 Concepts avancés

1. Asynchronisme .....	213
1.1 Fonctionnement de base .....	213
1.2 Thread et asynchronisme.....	215
1.3 Asynchronisme en C# .....	216
1.4 Les mots-clés async et await .....	218
1.5 Flux asynchrones .....	221

# 6 \_\_\_\_\_ C# 12 et Visual Studio Code

Les fondamentaux du langage

2. Algorithmique avancée.....	223
2.1 Programmation événementielle.....	223
2.1.1 Les delegates.....	223
2.1.2 Les events.....	225
2.2 Les types génériques.....	228
2.2.1 Utilisation standard.....	228
2.2.2 Contraintes sur type générique.....	230
2.3 Gestion de la mémoire.....	231
2.3.1 Le destructeur.....	232
2.3.2 IDisposable et IAsyncDisposable.....	233
2.4 Paramètres de méthodes avancés.....	235
2.4.1 Paramètre optionnel.....	235
2.4.2 Mots-clés de paramètres.....	235
2.4.3 Nommage de paramètres.....	238
2.4.4 Paramètres variables.....	239
2.5 Extension du fonctionnement d'un type.....	240
2.5.1 Méthodes d'extension.....	240
2.5.2 Définition des opérateurs.....	241
2.6 Tuples et déconstruction.....	245
2.6.1 Les tuples en C# 7.....	245
2.6.2 Déconstruction de type.....	248
2.7 Fonction locale.....	250

## Chapitre 8

### Créer des applications

1. Application web.....	253
1.1 Applications web graphiques.....	253
1.1.1 ASP.NET MVC.....	254
1.1.2 ASP.NET Razor Pages.....	259
1.1.3 Blazor.....	264
1.2 API.....	269

2. Application de bureau .....	275
2.1 WinForms .....	275
2.2 Windows Presentation Foundation (WPF) .....	281
2.3 Universal Windows Platform (UWP) .....	285
3. Application mobile .....	289
3.1 MAUI .....	290
3.2 Code .....	294
4. Conclusion .....	296

## **Chapitre 9**

### **Référence**

1. Introduction .....	297
2. Mots-clés de type .....	297
3. Mots-clés de programmation orientée objet .....	299
4. Mots-clés algorithmiques .....	303

Index .....	309
-------------	-----

Les éléments à télécharger sont disponibles à l'adresse suivante :  
**<http://www.editions-eni.fr>**  
Saisissez la référence ENI de l'ouvrage **RI4CAPOO** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1  
Introduction à la POO

1. Histoire de la POO ..... 15

2. Historique du C# ..... 18

Chapitre 2  
La conception orientée objet

1. Approche procédurale  
et décomposition fonctionnelle19

2. La transition vers l'approche objet. .... 20

3. Les caractéristiques de la POO. .... 21

3.1 L'objet, la classe et la référence ..... 21

3.1.1 L'objet ..... 21

3.1.2 La classe. .... 22

3.1.3 La référence ..... 23

3.2 L'encapsulation ..... 24

3.3 L'héritage. .... 24

3.4 Le polymorphisme ..... 26

3.5 L'abstraction ..... 27

4. Le développement objet ..... 28

4.1 Cahier des charges du logiciel ..... 28

4.2 Présentation du cycle en V. .... 28



# 2 \_\_\_\_\_ Apprendre la POO

avec le langage C#

4.3	Modélisation et représentation UML . . . . .	32
4.3.1	Diagrammes de cas d'utilisation . . . . .	35
4.3.2	Diagrammes de classes . . . . .	36
4.3.3	Énumérations . . . . .	42
4.3.4	Diagrammes de séquences . . . . .	43
4.4	Rédaction du code et des tests unitaires . . . . .	45
5.	Exercices corrigés . . . . .	46
5.1	Hiérarchie de classes . . . . .	46
5.2	Relations entre objets . . . . .	48
5.3	Agrégation d'objets . . . . .	48
5.4	Diagramme de cas d'utilisation . . . . .	50
5.5	Diagramme de séquences . . . . .	51

## Chapitre 3

### Introduction à .NET 6 et à VS

1.	Introduction . . . . .	53
2.	Environnement d'exécution . . . . .	54
3.	Le choix des langages . . . . .	55
4.	Utiliser plusieurs langages . . . . .	55
5.	Une librairie très complète . . . . .	55
6.	Un outil de développement complet . . . . .	57

## Chapitre 4

### Les types du C#

1.	"En C#, tout est typé !" . . . . .	67
2.	"Tout le monde hérite de System.Object" . . . . .	72
2.1	Les types Valeurs . . . . .	73
2.2	Les types Références . . . . .	76
2.3	Boxing/unboxing . . . . .	78

2.4	Utilisation des méthodes de System.Object . . . . .	79
2.4.1	Equals . . . . .	80
2.4.2	GetHashCode . . . . .	84
2.4.3	ToString . . . . .	86
2.4.4	Finalize . . . . .	87
2.4.5	Object.GetType et les opérateurs typeof et is . . . . .	88
2.4.6	object.ReferenceEquals. . . . .	89
2.4.7	Object.MemberwiseClone . . . . .	90
2.5	Le type System.String et son alias string . . . . .	93
3.	Exercice corrigé . . . . .	97
3.1	Énoncé . . . . .	97
3.2	Correction . . . . .	97

## Chapitre 5 Création d'objets

1.	Introduction . . . . .	101
2.	Les espaces de noms . . . . .	102
3.	Déclaration d'une classe . . . . .	110
3.1	Accessibilité des membres . . . . .	112
3.2	Attributs . . . . .	112
3.2.1	Attributs constants . . . . .	113
3.2.2	Attributs en lecture seule . . . . .	114
3.3	Propriétés. . . . .	116
3.4	Constructeur. . . . .	124
3.4.1	Étapes de la construction d'un objet . . . . .	124
3.4.2	Surcharge de constructeurs . . . . .	126
3.4.3	Constructeurs avec valeurs de paramètres par défaut . . . . .	126
3.4.4	Chaînage de constructeurs. . . . .	127
3.4.5	Les constructeurs de type static . . . . .	128
3.4.6	Les constructeurs de type private . . . . .	129
3.4.7	Les initialiseurs d'objets . . . . .	131

# 4 \_\_\_\_\_ Apprendre la POO

avec le langage C#

3.5	Destructeur . . . . .	132
3.6	Autre utilisation de using. . . . .	134
3.7	Le mot-clé this et ses vertus. . . . .	135
3.8	Méthodes . . . . .	138
3.8.1	Déclaration . . . . .	139
3.8.2	Passage par valeur et passage par référence. . . . .	142
3.9	Mécanisme des exceptions. . . . .	159
3.9.1	Présentation . . . . .	159
3.9.2	Principe de fonctionnement des exceptions . . . . .	160
3.9.3	Prise en charge de plusieurs exceptions . . . . .	169
3.9.4	try ... catch ... finally et using . . . . .	170
3.10	Surcharge des méthodes. . . . .	172
3.11	Exercice . . . . .	174
3.11.1	Énoncé . . . . .	174
3.11.2	Conseils . . . . .	175
3.11.3	Correction . . . . .	175
4.	Les interfaces. . . . .	178
4.1	Introduction . . . . .	178
4.2	Le contrat . . . . .	178
4.3	Déclaration d'une interface . . . . .	179
4.4	Implémentation . . . . .	181
4.5	Visual Studio et les interfaces . . . . .	183
4.6	Représentation UML d'une interface . . . . .	186
4.7	Interfaces et polymorphisme. . . . .	187
4.8	Exercice . . . . .	188
4.8.1	Énoncé . . . . .	188
4.8.2	Conseils . . . . .	188
4.8.3	Correction . . . . .	191
4.9	Les interfaces du .NET . . . . .	194

5. Association, composition et agrégation . . . . .	196
5.1 Les tableaux. . . . .	204
5.2 Les collections . . . . .	212
5.2.1 List<> et LinkedList<> . . . . .	213
5.2.2 Queue<T> et Stack<T> . . . . .	216
5.2.3 Dictionary<TKey, TValue> . . . . .	217
5.2.4 Les énumérateurs . . . . .	217
5.2.5 La magie du yield . . . . .	219
5.3 Exercice . . . . .	220
5.3.1 Énoncé. . . . .	220
5.3.2 Correction . . . . .	222
6. Les classes imbriquées. . . . .	223
7. Les structures . . . . .	225
7.1 Déclaration d'une structure . . . . .	226
7.2 Instanciation d'une structure. . . . .	228
8. Les classes partielles . . . . .	230
9. Les méthodes partielles. . . . .	231
10. Les indexeurs. . . . .	233
11. Surcharge d'opérateurs . . . . .	237
12. Fonctions locales. . . . .	240
13. Les objets "gourmands" en références faibles . . . . .	241
14. Les objets "dynamics" . . . . .	243
15. Les "Tuple" et "ValueTuple" . . . . .	244
16. Les records. . . . .	245
16.1 Introduction . . . . .	245
16.2 Déclaration complète . . . . .	246
16.3 Déclaration simplifiée. . . . .	247
16.4 Comparaison de records. . . . .	249
16.5 Déconstruction d'un record . . . . .	251
16.6 Mutation d'un record. . . . .	251

# 6 \_\_\_\_\_ Apprendre la POO

avec le langage C#

## Chapitre 6

### Héritage et polymorphisme

1. Comprendre l'héritage .....	253
2. Codage de la classe de base et de son héritière .....	254
2.1 Interdire l'héritage .....	254
2.2 Définir les membres héritables .....	255
2.3 Codage de l'héritage .....	255
2.4 Exploitation d'une classe héritière .....	256
3. Communication entre classe de base et classe héritière .....	257
3.1 Les constructeurs .....	257
3.2 Accès aux membres de base depuis l'héritier .....	260
3.3 Masquage ou substitution de membres hérités .....	262
3.3.1 Codage du masquage .....	264
3.3.2 Codage de la substitution .....	266
4. Exercice .....	267
4.1 Énoncé .....	267
4.2 Corrigé .....	268
5. Les classes abstraites .....	269
6. Les méthodes d'extension .....	271
7. Le polymorphisme .....	274
7.1 Comprendre le polymorphisme .....	274
7.2 Exploitation du polymorphisme .....	275
7.3 Les opérateurs is, as et () .....	275

## Chapitre 7

### Communication entre objets

1. L'événementiel : être à l'écoute .....	279
2. Le pattern Observateur .....	280

3.	La solution C# : delegate et event. . . . .	284
3.1	Utilisation du delegate dans le design pattern Observateur . .	287
3.2	Utilisation d'un event. . . . .	290
3.3	Comment accompagner l'event de données . . . . .	293
3.4	Les génériques en renfort pour encore simplifier . . . . .	295
3.5	Les expressions lambda . . . . .	296
3.6	Exemple d'utilisation d'event. . . . .	301
4.	Appels synchrones, appels asynchrones . . . . .	309
5.	Exercice . . . . .	311
5.1	Énoncé . . . . .	311
5.2	Conseils pour la réalisation . . . . .	312
5.3	Correction . . . . .	312
6.	Des messages entre les classes . . . . .	316

## Chapitre 8

### Le multithreading

1.	Introduction . . . . .	317
2.	Comprendre le multithreading . . . . .	318
3.	Multithreading et .NET . . . . .	321
4.	Implémentation en C# . . . . .	322
4.1	Utilisation d'un BackgroundWorker . . . . .	322
4.1.1	Communication du thread principal vers le thread secondaire . . . . .	324
4.1.2	Abandon du thread secondaire depuis le thread principal. . . . .	325
4.1.3	Communication du thread secondaire vers le thread principal . . . . .	326
4.1.4	Communication en fin de traitement du thread secondaire. . . . .	326
4.1.5	Exemple de code . . . . .	327
4.2	Utilisation du pool de threads créé par .NET . . . . .	329

# 8 --- Apprendre la POO

avec le langage C#

4.3	Gestion "manuelle" avec Thread/ParameterizedThreadStart . . .	331
5.	Synchronisation entre threads. . . . .	336
5.1	Nécessité de la synchronisation. . . . .	336
5.2	Le mot-clé lock . . . . .	338
5.3	La classe Monitor . . . . .	339
5.4	La classe Mutex. . . . .	340
5.5	La classe Semaphore . . . . .	341
6.	Communication entre threads . . . . .	342
6.1	Join . . . . .	342
6.2	Les synchronization events . . . . .	343
6.3	Communication entre threads secondaires et IHM. . . . .	350
6.4	Exercice . . . . .	353
6.4.1	Énoncé. . . . .	353
6.4.2	Correction . . . . .	353
7.	La programmation asynchrone . . . . .	357
7.1	Les « Task » . . . . .	357
7.2	async et await . . . . .	359
7.3	Le mot-clé async . . . . .	360
7.4	Contenu d'une méthode async . . . . .	360
7.5	Preuve à l'appui. . . . .	360
7.6	Retours possibles d'une méthode async . . . . .	362

## Chapitre 9

### P-Invoke

1.	Introduction . . . . .	365
1.1	Rappel sur les DLL non managées . . . . .	366
1.2	P-Invoke et son Marshal. . . . .	366
2.	Le cas simple . . . . .	367
2.1	Déclaration et appel . . . . .	368
2.2	Réglage de Visual Studio pour la mise au point . . . . .	370
3.	Appel avec paramètres et retour de fonction . . . . .	371

4. Traitement avec des chaînes de caractères . . . . .	373
4.1 Encodage des caractères . . . . .	373
4.2 Encodage des chaînes . . . . .	374
4.3 Transmission des chaînes. . . . .	375
5. Échange de tableaux . . . . .	378
5.1 Du C# au C/C++ . . . . .	378
5.2 Du C# au C/C++ puis retour au C# . . . . .	380
6. Partage de structures . . . . .	381
6.1 Déclaration des structures . . . . .	381
6.2 Utilisation des structures. . . . .	383
7. Les directives [In] et [Out]. . . . .	388
8. Réalisation d'un wrapper. . . . .	392
8.1 Une région "NativeMethods". . . . .	393
8.2 Stockage des informations de la DLL native. . . . .	394
8.3 Instanciation de DLL native. . . . .	395
8.4 Méthodes d'utilisation de la DLL managée depuis le wrapper . . . . .	397
8.5 Utilisation du wrapper. . . . .	398
9. Exercice . . . . .	399
9.1 Énoncé. . . . .	399
9.2 Correction. . . . .	400

## Chapitre 10 Les tests

1. Introduction . . . . .	403
2. Environnement d'exécution des tests unitaires . . . . .	405
3. Le projet de tests unitaires . . . . .	408
4. La classe de tests . . . . .	409
5. Contenu d'une méthode de test . . . . .	410
6. Traitements de préparation et de nettoyage . . . . .	413



# 10 \_\_\_\_\_ Apprendre la POO

avec le langage C#

7. DynamicData et source de données . . . . .	417
8. Automatisation des tests à la compilation . . . . .	422
9. Automatisation des tests en dehors de Visual Studio . . . . .	423
10. CodedUI . . . . .	425
11. Exercice . . . . .	426
11.1 Énoncé . . . . .	426
11.2 Correction . . . . .	426
12. Simulation par stub ou par shim . . . . .	428

## Chapitre 11

### Traçage et instrumentation des applications

1. Présentation . . . . .	431
2. Des objets de mise au point . . . . .	432
2.1 System.Diagnostics.Debug . . . . .	432
2.2 System.Diagnostics.Trace . . . . .	435
2.3 System.Diagnostics.TraceSource . . . . .	436
3. Principe de fonctionnement des écouteurs . . . . .	437
4. Comportement dynamique . . . . .	439
5. Mesurer le temps passé . . . . .	443
6. Exercice . . . . .	446
6.1 Énoncé . . . . .	446
6.2 Correction . . . . .	446

## Chapitre 12 La réflexion

1. Introduction .....	449
2. Mais pour quoi faire ? .....	450
3. Introspection d'une classe C# .....	452
3.1 Introspection "manuelle" .....	455
3.2 Introspection "logicielle" .....	458
3.2.1 Découverte et instanciation .....	458
3.2.2 Découverte et utilisation des propriétés .....	461
3.2.3 Découverte et utilisation des méthodes .....	463
3.3 Exercice .....	466
3.3.1 Énoncé .....	466
3.3.2 Quelques conseils .....	466
3.3.3 Correction .....	466
4. Chargement dynamique d'un objet implémentant une interface ..	469
4.1 Création d'une interface "plug-in" .....	470
4.2 Écriture d'un plug-in .....	471
4.3 L'application supportant les plug-ins .....	473
4.4 Exercice .....	475
4.4.1 Énoncé .....	475
4.4.2 Correction .....	475
5. Décompilation et obfuscation .....	477
6. Conclusion .....	483

## Chapitre 13

### Gestion des données

1. Introduction .....	485
2. LINQ .....	486
2.1 Qu'est-ce que c'est ? .....	486
2.2 Les deux syntaxes LINQ .....	487
2.2.1 La syntaxe "développeur SQL" .....	487
2.2.2 La syntaxe "développeur C#" .....	489
2.3 Requêtes et filtres .....	489
2.4 Quelques calculs .....	492
2.5 Regroupement des résultats .....	494
2.6 Les jointures .....	496
2.7 Exercice .....	498
2.7.1 Énoncé .....	498
2.7.2 Solution .....	499
3. Persistance des données en XML .....	506
3.1 Rappels sur le XML .....	507
3.2 XML et .NET .....	509
3.2.1 Sérialisation/désérialisation d'un modèle de données ..	509
3.2.2 Les décorations de sérialisation XML .....	509
3.2.3 XmlSerializer : écrire et lire .....	513
3.3 XSD.EXE, un outil de conversion .....	516
3.4 Exercice .....	518
3.4.1 Énoncé .....	518
3.4.2 Correction .....	518
3.5 LINQ to XML .....	520
3.5.1 Lecture .....	521
3.5.2 Écriture .....	522
3.5.3 Interrogations .....	524
3.6 Exercice .....	524
3.6.1 Énoncé .....	524
3.6.2 Correction .....	524

4.	Persistance dans des bases de données avec ADO.NET . . . . .	526
4.1	Présentation . . . . .	526
4.2	Les termes utilisés. . . . .	526
4.3	Les modules ADO.NET . . . . .	527
4.4	Notre environnement d'apprentissage . . . . .	528
4.5	ADO en mode connecté. . . . .	532
4.5.1	Les fournisseurs de données en .NET . . . . .	532
4.5.2	Se connecter avec DbConnection . . . . .	533
4.5.3	Envoyer des requêtes avec DbCommand . . . . .	537
4.5.4	Lire des enregistrements avec DbDataReader. . . . .	543
4.6	Exercice . . . . .	546
4.6.1	Énoncé. . . . .	546
4.6.2	Correction . . . . .	546
4.7	ADO en mode déconnecté. . . . .	548
4.7.1	La classe DataSet . . . . .	548
4.7.2	Le DataSet typé . . . . .	551
4.7.3	Persistance du DataSet en XML . . . . .	559
4.7.4	LINQ to DataSet . . . . .	561
4.7.5	Intégrité référentielle . . . . .	562
4.8	DbAdapter : jonction des deux modes . . . . .	569
4.8.1	Lecture de la source . . . . .	570
4.8.2	Mise à jour de la source . . . . .	573
5.	Entity Framework . . . . .	580
5.1	Présentation de l'Entity Data Model. . . . .	581
5.2	Création d'un EDM depuis une base de données. . . . .	582
5.3	DbContext . . . . .	588
5.4	LINQ to Entities . . . . .	594
5.5	Mise à jour de la source . . . . .	596
5.6	Création d'un EDM depuis un modèle . . . . .	600
5.7	Exercice . . . . .	612
5.7.1	Présentation du binding. . . . .	612
5.7.2	Énoncé. . . . .	613
5.7.3	Correction . . . . .	613

# 14 \_\_\_\_\_ Apprendre la POO

avec le langage C#

6. Conclusion . . . . .	614
-------------------------	-----

## Chapitre 14

### WPF MVVM et le toolkit Microsoft

1. Présentation . . . . .	615
2. Historique des API . . . . .	616
3. C# et XAML Développeur et Graphiste . . . . .	617
4. Balises et Attributs pour Objets et Propriétés . . . . .	617
5. Utilisation basique de WPF . . . . .	622
6. Utilisation des layout . . . . .	626
7. Récupération des informations SANS le binding . . . . .	633
8. Introduction au binding . . . . .	639
8.1 Le DataContext . . . . .	639
8.2 L'interface INotifyPropertyChanged . . . . .	642
8.3 Les convertisseurs . . . . .	645
8.4 Exercice . . . . .	648
8.5 Binding de commandes . . . . .	649
9. Le modèle de conception MVVM . . . . .	656
9.1 Objectifs . . . . .	656
9.2 Les dépendances . . . . .	657
9.3 Mise en application . . . . .	658
10. Présentation de MVVM Toolkit . . . . .	669
10.1 La classe Observable . . . . .	669
10.2 La classe ObservableValidator . . . . .	671
10.3 Messenger . . . . .	672
10.4 L'injection de dépendances . . . . .	675

Index . . . . .	681
-----------------	-----