

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI3CAPOO** 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 13
- 2. Historique du C# 16

Chapitre 2

La conception orientée objet

- 1. Approche procédurale et décomposition fonctionnelle 17
- 2. La transition vers l'approche objet 18
- 3. Les caractéristiques de la POO 19
 - 3.1 L'objet, la classe et la référence 19
 - 3.1.1 L'objet 19
 - 3.1.2 La classe 20
 - 3.1.3 La référence 21
 - 3.2 L'encapsulation 22
 - 3.3 L'héritage 22
 - 3.4 Le polymorphisme 24
 - 3.5 L'abstraction 25
- 4. Le développement objet 26
 - 4.1 Cahier des charges du logiciel 26
 - 4.2 Présentation du cycle en V 26

2 _____ Apprendre la POO

avec le langage C#

4.3	Modélisation et représentation UML	30
4.3.1	Diagrammes de cas d'utilisation	33
4.3.2	Diagrammes de classes	34
4.3.3	Énumérations	40
4.3.4	Diagrammes de séquences	41
4.4	Rédaction du code et des tests unitaires	43
5.	Exercices corrigés	44
5.1	Hiérarchie de classes	44
5.2	Relations entre objets	46
5.3	Agrégation d'objets	46
5.4	Diagramme de cas d'utilisation	48
5.5	Diagramme de séquences	49

Chapitre 3

Introduction au framework .NET et à VS

1.	Introduction	51
2.	Environnement d'exécution	51
3.	Le choix des langages	52
4.	Utiliser plusieurs langages	52
5.	Une librairie très complète	53
6.	Des installations simplifiées	53
7.	Un outil de développement complet	55

Chapitre 4

Les types du C#

1.	"En C#, tout est typé !"	65
2.	"Tout le monde hérite de System.Object"	70
2.1	Les types Valeurs	70
2.2	Les types Références	74

- 2.3 Boxing/unboxing 76
- 2.4 Utilisation des méthodes de System.Object 78
 - 2.4.1 Equals 78
 - 2.4.2 GetHashCode 82
 - 2.4.3 ToString 84
 - 2.4.4 Finalize 85
 - 2.4.5 Object.GetType et les opérateurs typeof et is 86
 - 2.4.6 Object.ReferenceEquals 87
 - 2.4.7 Object.MemberwiseClone 88
- 2.5 Le type System.String et son alias string 91
- 3. Exercice corrigé 95
 - 3.1 Énoncé 95
 - 3.2 Correction 95

Chapitre 5
Création de classes

- 1. Introduction 99
- 2. Les espaces de noms 99
- 3. Déclaration d'une classe 106
 - 3.1 Accessibilité des membres 108
 - 3.2 Attributs 108
 - 3.2.1 Attributs constants 109
 - 3.2.2 Attributs en lecture seule 110
 - 3.3 Propriétés 112
 - 3.4 Constructeur 120
 - 3.4.1 Étapes de la construction d'un objet 120
 - 3.4.2 Surcharge de constructeurs 122
 - 3.4.3 Constructeurs avec valeurs de paramètres par défaut 122
 - 3.4.4 Chaînage de constructeurs 123
 - 3.4.5 Les constructeurs de type static 124
 - 3.4.6 Les constructeurs de type private 125

4 --- Apprendre la POO

avec le langage C#

3.4.7	Les initialiseurs d'objets	127
3.5	Destructeur	128
3.6	Autre utilisation de using	130
3.7	Le mot-clé this et ses vertus	131
3.8	Méthodes	134
3.8.1	Déclaration	135
3.8.2	Passage par valeur et passage par référence	138
3.9	Mécanisme des exceptions	153
3.9.1	Présentation	153
3.9.2	Principe de fonctionnement des exceptions	155
3.9.3	Prise en charge de plusieurs exceptions	165
3.9.4	try ... catch ... finally et using	166
3.10	Surcharge des méthodes	169
3.11	Exercice	171
3.11.1	Énoncé	171
3.11.2	Conseils	172
3.11.3	Correction	173
4.	Les interfaces	175
4.1	Introduction	175
4.2	Le contrat	176
4.3	Déclaration d'une interface	177
4.4	Implémentation	178
4.5	Visual Studio et les interfaces	180
4.6	Représentation UML d'une interface	183
4.7	Interfaces et polymorphisme	184
4.8	Exercice	185
4.8.1	Énoncé	185
4.8.2	Conseils	185
4.8.3	Correction	188
4.9	Les interfaces du .NET	191
5.	Association, composition et agrégation	194
5.1	Les tableaux	201

5.2	Les collections	207
5.2.1	List<> et LinkedList<>	208
5.2.2	Queue<T> et Stack<T>	211
5.2.3	Dictionary<TKey, TValue>	212
5.2.4	Les énumérateurs	212
5.2.5	La magie du yield	214
5.3	Exercice	215
5.3.1	Énoncé	215
5.3.2	Correction	217
6.	Les classes imbriquées.	219
7.	Les structures	222
7.1	Déclaration d'une structure	222
7.2	Instanciation d'une structure.	224
8.	Les classes partielles	227
9.	Les méthodes partielles.	228
10.	Les indexeurs.	229
11.	Surcharge d'opérateurs	233
12.	Fonctions locales.	236
13.	Les objets "gourmands" en références faibles	237
14.	Les objets "dynamics"	239
15.	Les "Tuple" et "ValueTuple"	240

Chapitre 6

Héritage et polymorphisme

1.	Comprendre l'héritage	243
2.	Codage de la classe de base et de son héritière	244
2.1	Interdire l'héritage	244
2.2	Définir les membres héritables	245
2.3	Codage de l'héritage	245
2.4	Exploitation d'une classe héritée	246

6 --- Apprendre la POO

avec le langage C#

3.	Communication entre classe de base et classe héritière.	248
3.1	Les constructeurs	248
3.2	Accès aux membres de base depuis l'héritier	251
3.3	Masquage ou substitution de membres hérités	253
3.3.1	Codage du masquage	256
3.3.2	Codage de la substitution	258
4.	Exercice	259
4.1	Énoncé.	259
4.2	Corrigé.	260
5.	Les classes abstraites.	262
6.	Les méthodes d'extension.	263
7.	Le polymorphisme	267
7.1	Comprendre le polymorphisme.	267
7.2	Exploitation du polymorphisme	267
7.3	Les opérateurs is, as et ()	268

Chapitre 7

Communication entre objets

1.	L'événementiel : être à l'écoute	271
2.	Le pattern Observateur	272
3.	La solution C# : delegate et event.	276
3.1	Utilisation du delegate dans le design pattern Observateur	279
3.2	Utilisation d'un event.	282
3.3	Comment accompagner l'event de données	284
3.4	Les génériques en renfort pour encore simplifier	286
3.5	Les expressions lambda	287
3.6	Exemple d'utilisation d'event.	293
4.	Appels synchrones, appels asynchrones	301
4.1	Approche 1	303
4.2	Approche 2	304
4.3	Approche 3	305

- 4.4 Approche 3 avec une expression lambda 307
- 5. Exercice 309
 - 5.1 Énoncé 309
 - 5.2 Conseils pour la réalisation 310
 - 5.3 Correction 311

Chapitre 8
Le multithreading

- 1. Introduction 317
- 2. Comprendre le multithreading 317
- 3. Multithreading et .NET 320
- 4. Implémentation en C# 322
 - 4.1 Utilisation d'un BackgroundWorker 322
 - 4.1.1 Communication du thread principal
 vers le thread secondaire 325
 - 4.1.2 Abandon depuis le thread primaire 326
 - 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
 - 4.3 Gestion "manuelle" avec Thread/ParameterizedThreadStart . 332
- 5. Synchronisation entre threads 337
 - 5.1 Nécessité de la synchronisation 337
 - 5.2 La décoration Synchronization 340
 - 5.3 Le mot-clé lock 341
 - 5.4 La classe Monitor 343
 - 5.5 La classe Mutex 344
 - 5.6 La classe Semaphore 345

6. Communication entre threads	345
6.1 Join	345
6.2 Les synchronization events	347
6.3 Communication entre threads secondaires et IHM.	355
6.4 Exercice	358
6.4.1 Énoncé	358
6.4.2 Correction	359
7. La programmation asynchrone	363
7.1 Le mot-clé async	363
7.2 Contenu d'une méthode async	363
7.3 Preuve à l'appui.	364
7.4 Retours possibles d'une méthode async	365

Chapitre 9 P-Invoke

1. Introduction	369
1.1 Rappel sur les DLL non managées	370
1.2 P-Invoke et son Marshal.	370
2. Le cas simple	371
2.1 Déclaration et appel	372
2.2 Réglage de Visual Studio pour la mise au point	375
3. Appel avec paramètres et retour de fonction	376
4. Traitement avec des chaînes de caractères	378
4.1 Encodage des caractères	378
4.2 Encodage des chaînes	379
4.3 Transmission des chaînes.	379
5. Échange de tableaux	383
5.1 Du C# au C/C++	383
5.2 Du C# au C/C++ puis retour au C#	385

- 6. Partage de structures 387
 - 6.1 Déclaration des structures 387
 - 6.2 Utilisation des structures 389
- 7. Les directives [In] et [Out] 393
- 8. Réalisation d'un wrapper 397
 - 8.1 Une région "NativeMethods" 398
 - 8.2 Stockage des informations de la DLL native 398
 - 8.3 Instanciation de DLL native 399
 - 8.4 Méthodes d'utilisation de la DLL managée depuis le wrapper 402
 - 8.5 Utilisation du wrapper 402
- 9. Exercice 404
 - 9.1 Énoncé 404
 - 9.2 Correction 405

Chapitre 10
Les tests

- 1. Introduction 407
- 2. Environnement d'exécution des tests unitaires 409
- 3. Le projet de tests unitaires 412
- 4. La classe de tests 413
- 5. Contenu d'une méthode de test 414
- 6. Traitements de préparation et de nettoyage 417
- 7. TestContext et source de données 420
- 8. Automatisation des tests à la compilation 424
- 9. Automatisation des tests en dehors de Visual Studio 425
- 10. CodedUI 427
- 11. Exercice 428
 - 11.1 Énoncé 428
 - 11.2 Correction 428

Chapitre 11
La réflexion

1. Introduction	431
2. Mais pour quoi faire ?	431
3. Introspection d'une classe C#	434
3.1 Introspection "manuelle"	436
3.2 Introspection "logicielle"	439
3.2.1 Découverte et instanciation	439
3.2.2 Découverte et utilisation des propriétés	442
3.2.3 Découverte et utilisation des méthodes	444
3.3 Exercice	446
3.3.1 Énoncé	446
3.3.2 Quelques conseils	446
3.3.3 Correction	447
4. Chargement dynamique d'un objet implémentant une interface	450
4.1 Création d'une interface "plug-in"	450
4.2 Écriture d'un plug-in	451
4.3 L'application supportant les plug-ins	452
4.4 Exercice	455
4.4.1 Énoncé	455
4.4.2 Correction	455
5. Décompilation et obfuscation	457
6. Conclusion	463

Chapitre 12
Gestion des données

1. Introduction	465
2. LINQ	466
2.1 Qu'est-ce que c'est ?	466
2.2 Les deux syntaxes LINQ	467
2.2.1 La syntaxe "développeur SQL"	467

2.2.2	La syntaxe "développeur C#"	469
2.3	Requêtes et filtres	469
2.4	Quelques calculs	472
2.5	Regroupement des résultats	473
2.6	Les jointures	475
2.7	Exercice	477
2.7.1	Énoncé	477
2.7.2	Solution	478
3.	Persistance des données en XML	485
3.1	Rappels sur le XML	486
3.2	XML et .NET	488
3.2.1	Sérialisation/désérialisation d'un modèle de données	488
3.2.2	Les décorations de sérialisation XML	488
3.2.3	XmlSerializer : écrire et lire	492
3.3	XSD.EXE, un outil de conversion	495
3.4	Exercice	496
3.4.1	Énoncé	496
3.4.2	Correction	497
3.5	LINQ to XML	499
3.5.1	Lecture	499
3.5.2	Écriture	501
3.5.3	Interrogations	502
3.6	Exercice	503
3.6.1	Énoncé	503
3.6.2	Correction	503
4.	Persistance dans des bases de données avec ADO.NET	504
4.1	Présentation	504
4.2	Les termes utilisés	505
4.3	Les modules ADO.NET	506
4.4	Notre environnement d'apprentissage	507
4.5	ADO en mode connecté	510
4.5.1	Les fournisseurs de données en .NET	510
4.5.2	Se connecter avec DbConnection	511

12 --- Apprendre la POO

avec le langage C#

4.5.3	Envoyer des requêtes avec DbCommand	515
4.5.4	Lire des enregistrements avec DbDataReader.	521
4.6	Exercice	524
4.6.1	Énoncé.	524
4.6.2	Correction	524
4.7	ADO en mode déconnecté.	526
4.7.1	La classe DataSet	526
4.7.2	Le DataSet typé	529
4.7.3	Persistance du DataSet en XML	537
4.7.4	LINQ to DataSet	539
4.7.5	Intégrité référentielle	539
4.8	DbAdapter : jonction des deux modes	546
4.8.1	Lecture de la source	546
4.8.2	Mise à jour de la source	549
5.	Entity Framework	556
5.1	Présentation de l'Entity Data Model.	557
5.2	Création d'un EDM depuis une base de données.	558
5.3	DbContext	568
5.4	LINQ to Entities	570
5.5	Mise à jour de la source	572
5.6	Création d'un EDM depuis un modèle	575
5.7	Exercice	586
5.7.1	Présentation du binding.	586
5.7.2	Énoncé.	587
5.7.3	Correction	588
6.	Conclusion	589
	Index	591

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI19CSHA** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1

La plateforme .NET

- 1. Introduction 15
- 2. Historique de la plateforme 17
- 3. Évolution de la plateforme 25
 - 3.1 .NET Standard 25
 - 3.2 .NET Compiler Platform : Roslyn 26
 - 3.3 .NET dans le monde open source 27
- 4. Le Common Language Runtime (CLR) 28
- 5. La Base Class Library (BCL) 31
- 6. Le Dynamic Language Runtime (DLR) 32
- 7. Une première application avec C# 33
 - 7.1 Création 33
 - 7.2 Compilation 37
 - 7.3 Analyse de l'assembly 41
 - 7.3.1 Structure 41
 - 7.3.2 Exploration avec ILDASM 42

2 _____ C# 8 et Visual Studio 2019

Les fondamentaux du langage

Chapitre 2 Visual Studio 2019

1. Installation et premier lancement	47
1.1 Prérequis	47
1.2 Éditions de Visual Studio	48
1.2.1 Visual Studio Community	48
1.2.2 Éditions commerciales	49
1.3 Installation	50
1.4 Premier lancement	56
2. Description des outils	64
2.1 Barres d'outils	68
2.2 Explorateur de solutions	70
2.3 Explorateur d'objets	71
2.4 Explorateur de serveurs	72
2.5 Fenêtre de propriétés	76
2.6 Fenêtre d'édition de code	79
2.6.1 Navigation	79
2.6.2 Suivi des modifications	79
2.6.3 Mise en surbrillance des références	80
2.6.4 Refactorisation	81
2.6.5 IntelliSense	83
2.6.6 Snippets	83

Chapitre 3 L'organisation d'une application

1. Les solutions	85
1.1 Présentation	85
1.2 Création d'une solution	86
1.3 Organisation	87
1.4 Actions disponibles sur une solution	88
1.4.1 Ajout et suppression d'un projet	88
1.4.2 Création d'un dossier de solution	89

- 1.4.3 Chargement et déchargement d'un projet 90
- 1.4.4 Création d'un fichier 91
- 1.4.5 Génération de la solution 91
- 1.5 Configuration de la solution 92
 - 1.5.1 Configuration des projets de démarrage 92
 - 1.5.2 Dépendances du projet 94
 - 1.5.3 Paramètres d'analyse du code 95
 - 1.5.4 Fichiers sources pour le débogage 95
 - 1.5.5 Configuration 96
- 2. Les projets 97
 - 2.1 Création d'un projet 97
 - 2.2 Propriétés d'un projet 103
 - 2.2.1 Application 103
 - 2.2.2 Build 105
 - 2.2.3 Événements de build 109
 - 2.2.4 Package 110
 - 2.2.5 Déboguer 111
 - 2.2.6 Ressources 113
 - 2.2.7 Paramètres 114

Chapitre 4
Les bases du langage

- 1. Introduction 117
- 2. Les variables 117
 - 2.1 Nommage des variables 118
 - 2.2 Type des variables 119
 - 2.2.1 Types valeurs et types références 119
 - 2.2.2 Types intégrés 120
 - 2.3 Déclaration des variables 124
 - 2.4 Portée des variables 125
 - 2.5 Modificateurs d'accès 126
 - 2.6 Le mot-clé var et l'inférence de type 126

4 _____ C# 8 et Visual Studio 2019

Les fondamentaux du langage

3.	Les constantes	128
4.	Les opérateurs	128
4.1	Les opérateurs d'accès	129
4.1.1	Accès simple :	129
4.1.2	Accès indexé : []	129
4.1.3	Accès avec nullité conditionnelle : ?	130
4.2	Les opérateurs arithmétiques	131
4.3	Les opérateurs de comparaison	131
4.4	Les opérateurs conditionnels	132
4.4.1	Opérateur ternaire : ? ... :	132
4.4.2	Opérateur de fusion de valeur nulle : ??	133
4.5	Les opérateurs logiques	133
4.5.1	Négation : !	133
4.5.2	ET logique : &	134
4.5.3	OU logique : 	134
4.5.4	OU exclusif : ^	134
4.5.5	ET conditionnel : &&	135
4.5.6	OU conditionnel : 	135
4.6	Les opérateurs binaires	136
4.6.1	ET binaire : &	136
4.6.2	OU binaire : 	136
4.6.3	OU exclusif : ^	137
4.6.4	Négation : ~	137
4.6.5	Décalage vers la droite : >>	137
4.6.6	Décalage vers la gauche : <<	138
5.	Les structures de contrôle	138
5.1	Les structures conditionnelles	138
5.1.1	if ... else	138
5.1.2	switch	140
5.1.3	Le pattern matching	141
5.2	Les structures d'itération	146
5.2.1	for	146
5.2.2	while	147

- 5.2.3 do ... while 147
- 5.2.4 foreach 147
- 5.2.5 Contrôler l'exécution d'une boucle 148
- 5.3 Autres structures 149
 - 5.3.1 using 149
 - 5.3.2 goto 150
- 6. Les fonctions 151
 - 6.1 Écriture d'une fonction 151
 - 6.2 Paramètres de fonction 153
 - 6.3 Procédures 158
 - 6.4 Surcharges 158
 - 6.5 Fonctions locales 160
- 7. Les tuples 161
- 8. Les attributs 164

Chapitre 5

La programmation orientée objet avec C#

- 1. Les principes de la programmation orientée objet 165
- 2. Les classes et les structures 168
 - 2.1 Classes 168
 - 2.1.1 Déclaration 169
 - 2.1.2 Constructeur et destructeur 171
 - 2.1.3 Classes partielles 175
 - 2.2 Structures 176
 - 2.3 Création de méthodes 177
 - 2.3.1 Création 177
 - 2.3.2 Méthodes partielles 179
 - 2.3.3 Méthodes d'extension 180
 - 2.3.4 Méthodes opérateurs 182
 - 2.4 Création de propriétés 184
 - 2.4.1 Lecture et écriture 184

6 _____ C# 8 et Visual Studio 2019

Les fondamentaux du langage

2.4.2	Lecture seule	185
2.4.3	Écriture seule	186
2.4.4	Propriétés automatiques	186
2.4.5	Initialisation de propriétés automatiques	187
2.4.6	Propriétés automatiques en lecture seule	188
2.4.7	Propriétés indexées	188
2.5	Membres statiques	189
2.6	Utilisation des classes et structures	190
2.6.1	Instanciation	190
2.6.2	Initialisation	191
2.6.3	Types anonymes	192
3.	Les espaces de noms	195
3.1	Nomenclature	195
3.2	using	196
4.	L'héritage	198
4.1	Mise en œuvre	198
4.2	Les mots-clés this et base	199
4.3	Redéfinition et masquage	200
4.3.1	Redéfinition de méthode	201
4.3.2	Masquage de méthode	202
4.3.3	Différences entre redéfinition et masquage	202
4.4	Imposer ou interdire l'héritage	205
4.5	Le transtypage	206
5.	Les interfaces	208
5.1	Création	209
5.2	Utilisation	211
5.2.1	Implémentation implicite	212
5.2.2	Implémentation explicite	214
6.	Les énumérations	216
7.	Les délégués	216
7.1	Création	217
7.2	Utilisation	217

7.3	Expressions lambda	218
8.	Les événements	219
8.1	Déclaration et déclenchement	219
8.2	Gestion des événements	221
9.	Les génériques	222
9.1	Classes	223
9.1.1	Définition d'une classe générique	223
9.1.2	Utilisation d'une classe générique	224
9.2	Interfaces	224
9.2.1	Définition d'une interface générique	225
9.2.2	Utilisation d'une interface générique	226
9.3	Contraintes	227
9.4	Méthodes	232
9.4.1	Définition d'une méthode générique	232
9.4.2	Utilisation d'une méthode générique	234
9.5	Événements et délégués	234
10.	Les collections	236
10.1	Types existants	236
10.1.1	Array	237
10.1.2	ArrayList et List<T>	237
10.1.3	Hashtable et Dictionary<TKey, TValue>	241
10.1.4	Stack et Stack<T>	244
10.1.5	Queue et Queue<T>	244
10.2	Choisir un type de collection	245
11.	La programmation dynamique	245
12.	La programmation asynchrone	248
12.1	Les objets Task	248
12.2	Écrire du code asynchrone avec async et await	251

Chapitre 6

Le débogage et la gestion des erreurs

1. Les différents types d'erreurs	253
1.1 Erreurs de compilation	253
1.2 Erreurs d'exécution	255
2. Utilisation des exceptions	256
2.1 Création et déclenchement d'exceptions	256
2.1.1 La classe Exception	256
2.1.2 Le mot-clé throw	257
2.1.3 Exceptions spécialisées	257
2.2 Gérer les exceptions	258
2.2.1 La structure try ... catch	258
2.2.2 Les filtres d'exception	261
2.2.3 Le bloc finally	262
3. Les outils fournis par Visual Studio	264
3.1 Contrôle de l'exécution	264
3.1.1 Démarrage	265
3.1.2 Arrêt	266
3.1.3 Pause	266
3.1.4 Reprise	267
3.2 Points d'arrêt	267
3.3 Visualiser le contenu des variables	274
3.3.1 DataTips	274
3.3.2 Fenêtres Espion	275
3.3.3 Fenêtre Espion express	276
3.3.4 Fenêtre Automatique	277
3.3.5 Fenêtre Variables locales	277
3.4 Compilation conditionnelle	277
4. L'erreur à un milliard de dollars : les références nulles	280
4.1 NullReferenceException	281
4.2 Une solution : l'utilisation de types références non-nuls	283

Chapitre 7**Le développement d'applications Windows**

1. Présentation de WPF.	289
1.1 Structure d'une application WPF.	290
1.2 XAML	291
1.2.1 Templates	292
1.2.2 Espaces de noms	293
1.3 Contexte de données et binding	294
2. Utilisation des contrôles.	298
2.1 Ajout de contrôles	299
2.2 Positionnement et dimensionnement des contrôles	302
2.3 Ajout d'un gestionnaire d'événements à un contrôle	305
3. Les principaux contrôles.	307
3.1 Contrôles de fenêtrage	308
3.1.1 Window.	308
3.1.2 NavigationWindow	311
3.2 Contrôles de disposition.	313
3.2.1 Grid	313
3.2.2 StackPanel	316
3.2.3 DockPanel	317
3.2.4 WrapPanel	319
3.2.5 Canvas.	320
3.3 Contrôles d'affichage de données	321
3.3.1 TextBlock	321
3.3.2 Label	322
3.3.3 Image.	324
3.3.4 ScrollViewer	325
3.3.5 ItemsControl.	325
3.3.6 StatusBar.	329
3.3.7 ToolTip	329

3.4	Contrôles d'édition de texte.....	330
3.4.1	TextBox.....	330
3.4.2	RichTextBox.....	331
3.4.3	PasswordBox.....	333
3.5	Contrôles de sélection.....	334
3.5.1	RadioButton.....	334
3.5.2	CheckBox.....	334
3.5.3	ComboBox.....	335
3.5.4	ListBox.....	336
3.5.5	ListView.....	337
3.5.6	TreeView.....	340
3.5.7	Slider.....	345
3.5.8	Calendar.....	345
3.5.9	DatePicker.....	346
3.6	Contrôles d'action.....	346
3.6.1	Button.....	347
3.6.2	Menu.....	347
3.6.3	ContextMenu.....	350
3.6.4	ToolBar.....	350
4.	Interactions clavier et souris.....	352
4.1	Événements clavier.....	352
4.2	Événements souris.....	355
4.3	Glisser-déposer.....	357
5.	Aller plus loin avec WPF.....	360
5.1	Introduction à l'utilisation de Blend.....	360
5.1.1	L'interface.....	361
5.1.2	Ajout et modification de contrôles visuels.....	368
5.2	Introduction à MVVM.....	371
5.2.1	Présentation.....	371
5.2.2	Les interfaces INotifyPropertyChanged et INotifyCollectionChanged.....	372
5.2.3	Commandes.....	373
5.2.4	Mise en œuvre.....	373

Chapitre 8
L'accès aux données

- 1. Principes d'une base de données 389
 - 1.1 Terminologie..... 389
 - 1.2 Le langage SQL 390
 - 1.2.1 Recherche d'enregistrements 391
 - 1.2.2 Ajout d'enregistrements..... 393
 - 1.2.3 Mise à jour d'informations..... 393
 - 1.2.4 Suppression d'informations 394
- 2. ADO.NET 394
 - 2.1 Présentation 394
 - 2.2 Les fournisseurs de données..... 395
 - 2.2.1 SQL Server..... 396
 - 2.2.2 OLE DB 396
 - 2.2.3 ODBC 397
- 3. Utiliser ADO.NET en mode connecté..... 397
 - 3.1 Connexion à une base de données..... 398
 - 3.1.1 Chaînes de connexion 398
 - 3.1.2 Pools de connexions 401
 - 3.1.3 Gestion de la connexion..... 403
 - 3.2 Création et exécution de commandes..... 405
 - 3.2.1 Définition et création d'une commande 406
 - 3.2.2 Sélection de données 406
 - 3.2.3 Actions sur les données 408
 - 3.2.4 Paramétrage d'une commande 408
 - 3.2.5 Exécution de procédures stockées 411
- 4. Utiliser ADO.NET en mode déconnecté..... 413
 - 4.1 DataSet et DataTable..... 413
 - 4.1.1 Description 413
 - 4.1.2 Remplissage d'un DataSet
à partir d'une base de données 414
 - 4.1.3 Remplissage d'un DataSet sans base de données 417

4.2	Manipulation des données hors connexion	420
4.2.1	Lecture des données	421
4.2.2	Création de contraintes	421
4.2.3	Relations entre DataTables	425
4.2.4	État et versions d'une DataRow	427
4.2.5	Modification de données	428
4.2.6	Suppression de données	430
4.2.7	Valider ou annuler des modifications	430
4.2.8	Filtrage et tri à l'aide d'une DataView	431
4.2.9	Recherche de données	434
4.3	Valider les modifications au niveau de la base de données	436
4.3.1	Générer des commandes de mise à jour automatiquement	437
4.3.2	Commandes de mise à jour personnalisées	439
4.3.3	Gestion des accès concurrentiels	440
5.	Utiliser les transactions	442

Chapitre 9

LINQ

1.	Présentation de LINQ	445
2.	Syntaxe	446
2.1	Une première requête LINQ	449
2.2	Les opérateurs de requête	452
2.2.1	Projection	452
2.2.2	Filtrage	454
2.2.3	Triage	456
2.2.4	Partitionnement	457
2.2.5	Jointure et regroupement	458
2.2.6	Agrégation	461
3.	Entity Framework Core	463
3.1	Les principes du mappage objet-relationnel	463
3.1.1	Les principes	463

- 3.2 Mise en place. 465
 - 3.2.1 Génération du code à partir d'une base de données . . . 467
 - 3.2.2 Création d'une base de données avec du code 481
- 3.3 Utilisation de LINQ avec Entity Framework Core. 483
 - 3.3.1 Récupération de données 484
 - 3.3.2 Mise à jour de données. 486
 - 3.3.3 Gestion des conflits 488

Chapitre 10

XML

- 1. Présentation 493
- 2. Structure d'un fichier XML 494
 - 2.1 Constituants d'un document XML 494
 - 2.2 Document bien formé et document valide 498
- 3. Manipuler un document XML. 499
 - 3.1 Utilisation de DOM 500
 - 3.2 Utilisation de XPath 505
 - 3.3 Utilisation de LINQ to XML 508

Chapitre 11

Le déploiement

- 1. Introduction 513
- 2. Windows Installer 514
 - 2.1 Création d'un projet d'installation 515
 - 2.1.1 Opérations sur le système de fichiers 517
 - 2.1.2 Opérations sur le registre de la machine cible 521
 - 2.1.3 Configuration de l'installation par l'utilisateur 522
 - 2.1.4 Exécution d'actions personnalisées 524
 - 2.1.5 Conditions 525
- 3. Fichiers binaires (.exe, .dll). 527

14 _____ C# 8 et Visual Studio 2019

Les fondamentaux du langage

4. ClickOnce	533
4.1 La technologie ClickOnce	533
4.1.1 Principes de fonctionnement	534
4.1.2 Méthodes de déploiement disponibles	535
4.1.3 Les mises à jour d'applications avec ClickOnce	536
4.2 La publication ClickOnce	538
5. MSIX	547

Chapitre 12

Aide-mémoire

1. Introduction	563
Index	583