

Secteur Tertiaire Informatique
Filière « Etude et développement »

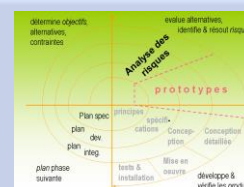
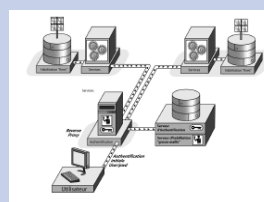
Construire une application organisée en couches
en mettant en œuvre des Frameworks

Manipuler des objets avec Linq

Apprentissage

Mise en pratique

Evaluation



Version	Date	Auteur(s)	Action(s)
1.0	07/01/20	Vincent Bost	Création du document

TABLE DES MATIERES

1. EXPRESSIONS LAMBDA ET METHODES EXTENSION DE TYPE	5
2. INTERROGER VOS DONNEES.....	5
2.1.1 Manipulation de valeurs numériques	5
2.1.2 Manipulation de types complexes	6
3. APPLICATION PRATIQUE : GERER SERVICES	6
4. APPLICATION PRATIQUE : RECHERCHE DOCUMENT	7

Préambule

Ce document propose une série d'exercices autour de la mise en place d'un Framework de persistance.

Pour réaliser ces exercices, il vous faut disposer :

Framework ORM :

- Entity Framework Core 3.0

Pour la manipulation des objets :

- Le langage de requête Linq

Objectifs

A l'issue de la réalisation de ces exercices, vous serez capable de concevoir et programmer :

- Des expressions lambda
- Des extensions de classe
- Des requêtes Linq sur des objets

Méthodologie

Nous allons aborder ici la manipulation des objets en mémoire sans nous soucier, dans un premier temps, des aspects spécifiques aux ORM. En effet, ici, pas de couplage avec un système de gestion de base de données relationnel.

Cette première phase vous facilitera l'apprentissage des requêtes Linq ciblant des bases de données relationnelles.

Références utiles pour la réalisation de ces exercices pratiques :

- A-ORM-Manipulation Objets : Manipuler les objets
- <https://www.tutorialsteacher.com/linq/sample-linq-queries>

Confrontez vos travaux à ceux de vos pairs et essayez de résoudre vos difficultés par l'échange avec ceux-ci ou votre formateur avant de prendre connaissance de la solution lorsque celle-ci est mise à votre disposition.

1. EXPRESSIONS LAMBDA ET METHODES EXTENSION DE TYPE

Tous ces exercices sont réalisés au sein d'un projet de type Application Console Core.

1. Créez une fonction à l'aide d'une expression lambda qui retourne la somme des longueurs de deux chaînes fournies en argument.
2. Créez une méthode d'extension WordsCount de la classe String qui permet de calculer le nombre de mots dans une chaîne. Je vous suggère d'utiliser split avec un tableau de séparateur.
3. Créez une méthode d'extension de IEnumerable<Double> Median qui permet de calculer la valeur médiane d'une série de valeurs numériques. Pour rappel, les valeurs de la série sont classées dans l'ordre croissant : Si le nombre de valeurs est impair, la médiane est la valeur du milieu. S'il est pair, la médiane est la somme des deux valeurs du milieu / 2.
4. Testez les méthodes d'extension créées.

Pour exemple, produire une séquence des chaînes dont le nombre de mots est supérieur à une valeur donnée.

Calculez la valeur médiane d'une série avec nombre de valeurs paire

Calculez la valeur médiane d'une série avec nombre de valeurs impair

Vérifier la robustesse de votre méthode avec une série comportant aucun élément et une série comportant 1 élément.

2. INTERROGER VOS DONNEES

Vous allez poursuivre ici votre apprentissage de Linq avec la formulation de requêtes d'interrogation de données.

Vous allez découvrir dans les exercices qui suivent les techniques :

- de projection,
- de filtres,
- de regroupement,
- de partitionnement,
- d'agrégation,
- et de jointure.

Vous utiliserez la syntaxe des requêtes ou celle des méthodes selon votre préférence. N'hésitez pas à aborder les deux formes si vous disposez de temps suffisant !

Il peut exister plusieurs formes d'expression de requêtes pour un résultat final identique mais au prix de performances très différentes. N'hésitez pas à intégrer cette problématique dans l'expression de vos requêtes....

Afin d'avoir des résultats qui puissent être comparables, ajoutez la classe JeuEssai.cs fournie à un projet console core .

2.1.1 Manipulation de valeurs numériques

Créez un tableau de valeurs numériques à partir de la méthode **CreeListeEntiers** de JeuEssai. Ces éléments se trouvent dans le fichier **LinqToObject_A1.zip**.

5. Afficher les valeurs Min et Max
6. Afficher la moyenne

Manipulation d'objets avec Linq

Afpa © 2020 – Section Tertiaire Informatique – Filière « Etude et développement »

7. Réaliser la somme des nombres
8. Variante 3 : Pour tout nombre x de la liste, faire la somme de $x * 2$ si x est pair, de $x * 3$ si x est impair.
9. Ecrire la liste des nombres ordonnés
10. Donner la liste des nombres dont le carré est supérieur à 5
11. Donner la liste des nombres divisibles par 5
12. Donner la liste des nombres dont le carré est supérieur au carré de la moyenne
13. Créer une nouvelle liste de nombres entiers positifs contigus de 1 à 5 avec la méthode range. Calculer le factoriel de 5. Pour rappel $1*2*3*4*5$. Le résultat est 120.

2.1.2 Manipulation de types complexes

La définition du type Stagiaire vous est fournie dans la classe `JeuEssai.cs` qui se trouve dans **LinqToObject_A1.zip**.

Créez la liste de stagiaires à l'aide de la méthode `CreeListeStagiaire`

1. Trouver le plus jeune et le plus vieux des stagiaires
2. Trouver celui qui a le nom le plus long
3. Donner une liste ordonnée en fonction du nombre de caractères du nom
4. Ordonner la liste des stagiaires selon leur âge, du plus âgé au plus jeune.
5. Donner la liste de ceux dont le nom commence par une suite de caractères
6. Donner la liste des stagiaires dont le nom contient une suite de caractères.
7. Donner la liste des stagiaires dont le rang, calculé en fonction du nom, est compris entre 3 et 6.
8. Donner l'âge moyen des stagiaires
9. Donner la liste des stagiaires dont l'âge est supérieur à l'âge moyen.

3. APPLICATION PRATIQUE : GERER SERVICES

Vous devez programmer une application qui permettra d'arrêter le service sélectionné parmi la liste de ceux en cours d'exécution sur votre machine. Cette application sera de type console.

Les informations suivantes vous sont communiquées afin de vous aider à mener à bien votre tâche.

Pour accéder aux services qui s'exécutent sur votre machine, ajouter une référence à la bibliothèque **system.serviceprocess**

La méthode statique **GetServices** de la classe **ServiceController** vous permet d'extraire la liste des services. Reportez-vous à la documentation msdn de la classe `ServiceController`.

L'application qui demande l'arrêt d'un service doit s'exécuter dans un contexte de sécurité adapté. Le plus simple est ici d'exécuter l'application en tant qu'administrateur.

Lors de l'arrêt d'un service, les services qui dépendent de celui-ci seront aussi arrêtés.

Un service actif possède l'état **running**.

L'identification d'un service sera effectuée par son nom.

Ecrire la requête linq qui permet d'arrêter, parmi les services actifs, ceux dont le nom contient la valeur passée en paramètre (exemple MSSQLSER).

4. APPLICATION PRATIQUE : RECHERCHE DOCUMENT

S'il vous reste du temps...

Je ne sais pas si vous rencontrez le même souci que moi, mais le trouve que la recherche de document dans les dossiers n'est pas très pratique.

Pouvons-nous envisager un mécanisme qui rechercherait un document à partir d'un répertoire racine et dans l'ensemble des sous-répertoires.

- Avec son type (extension)
- Un élément recherché dans le nom du fichier
- Puis éventuellement la date de dernière modification

Et pour les fichiers de type dans un format text éventuellement avec une recherche dans le contenu.

Voire aussi pour les documents Word qui ont une partie, dans le zip, qui comporte les éléments de contenu.