

WORKSHOP

Tout afficher + Tout réduire -

INTRODUCTION

Avant de commencer, bien lire le fichier fourni ci-dessous !

Pré requis - Workshop_Conception et Modèles de Programmation.pdf [pdf] ⬇

SUJET

✓ Énoncé

EXERCICE : CONCEPTION D'UNE ARCHITECTURE D'APPLICATION DE TYPE MVVM

Introduction

Pour ce workshop, vous allez développer en C# une petite application de conversion d'une chaîne de caractères de minuscules à majuscules. Cette application devra être conçue selon un schéma d'architecture de type MVVM et pourra évoluer fonctionnellement en limitant au maximum la propagation des modifications dans le code source.


Pour cela, vous allez utiliser l'environnement de développement logiciel que vous avez installé et configuré pendant la séquence 0. Cet environnement de travail est composé de :

- Visual Studio
- .NET Core 3.0 ou plus
- Git

Vous n'oublierez donc pas de réaliser les activités proposées dans ce workshop en utilisant l'outil Git de gestion des versions que vous avez installé dans votre environnement.

De plus, vous réaliserez la conception de votre application avec le formalisme UML que vous utiliserez pour décrire, **avant de coder**, votre modèle d'architecture MVVM ainsi que les objets et données qui le composent. Vous proposerez au moins une vue statique (diagramme de classes) et au moins une vue dynamique (diagramme de séquences) de votre architecture.

Si besoin, l'outil gratuit ArgoUML pourra vous aider pour travailler la conception UML de votre application.

<https://www.clubic.com/telecharger-fiche122322-argouml.html> 

Nota : Dans le cas d'un choix d'utilisation de ArgoUML, il est conseillé d'avoir d'abord installé en pré-requis la dernière version disponible du JRE pour Windows. En effet, les installations du JRE et de ArgoUML au travers de l'installateur de ArgoUML ne donnent pas entière satisfaction.

<https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html> 

Première version de votre application - Le mode console

La première version de votre application de conversion en majuscules d'une chaîne de caractères quelconque (voir exigences ci-dessous) répond aux exigences suivantes.

Les interactions avec l'utilisateur (saisie des données d'entrée, affichage des données de sortie) s'effectuent en mode console.

D'autre part,

- L'utilisateur saisit une chaîne de caractères dont le nombre de caractères varie entre 1 et 8.
- L'application affiche en majuscule la chaîne de caractères saisie par l'utilisateur.
- Seules les lettres de l'alphabet saisies en minuscules sont converties en majuscules.
- Les lettres saisies en majuscules sont reportées sans traitement dans la chaîne de caractères à afficher.
- Les caractères alphanumériques sont reportés sans traitement dans la chaîne de caractères à afficher

- Les caractères spéciaux &, % ... sont reportés sans traitement dans la chaîne de caractères à afficher.
- Une fois la chaîne de caractères affichée en majuscule, la fenêtre console reste affichée tant qu'une touche clavier n'est pas activée par l'utilisateur.

Pour cette première version de votre application en mode console, vous devez produire une conception UML de l'architecture et des données de votre application.

1. L'architecture doit suivre un modèle d'architecture d'application de type MVVM
2. Votre conception UML doit présenter, à minima, un diagramme de séquence (vue dynamique) et un diagramme de classe (vue statique).

Une fois la conception terminée, vous procédez au développement en langage C# de votre application en mode console.

Avant de considérer le développement de votre application en mode console comme TERMINE, vous procéderez aux tests de recette suivants :

| Test | Chaîne saisie | Chaîne affichée | Statut Test (OK/KO) | Commentaire |
|------|---------------|-----------------|---------------------|-------------|
| 1 | maison | MAISON | OK | |
| 3 | Corse123 | CORSE123 | OK | |
| 4 | %&ciel | %&CIEL | OK | |
| 5 | !*port | !*PORT | OK | |

Dès lors que les tests de recette sont tous OK, vous pouvez passer au développement de la deuxième version de votre application.

Deuxième version de votre application - Le mode graphique

Cette deuxième version de votre application répond au même cahier des charges de conversion, en majuscules, d'une chaîne de caractères en minuscules.

L'évolution fonctionnelle par rapport à la première version consiste à gérer les interactions avec l'utilisateur en mode graphique au travers d'une IHM (Interface Homme Machine).

Le développement de cette deuxième version de votre application devra s'appuyer sur WPF (Windows Presentation Foundation), les spécifications graphiques du framework .NET.

L'interface graphique devra comporter :

- Un champ éditable pour la saisie de la chaîne à convertir.
- Un champ non éditable pour l'affichage de la chaîne convertie en majuscule.
- Un bouton poussoir pour lancer le traitement de conversion.

Pour cette deuxième version de votre application en mode graphique, vous devez faire évoluer votre conception UML de l'architecture et des données de l'application précédemment conçue pour le mode console.

Vous montrerez et commenterez alors comment l'architecture de type MVVM permet de limiter la propagation des modifications du code source pour passer de la première version (mode console) à la deuxième version (mode graphique).

Une fois la conception terminée, vous procéderez au développement en langage C# de votre application en mode graphique.

Bien entendu, avant de considérer le développement de votre application en mode graphique comme TERMINE, vous procédez de nouveau aux tests de recette tels que proposés pour le mode console. Vous vérifierez alors que tous les tests de recette sont OK.

Bon travail !

