

Mission 2 : Gestion de la clôture

Difficulté	Facile (sauf dernière tâche : moyenne)
Temps estimé	20h
Prérequis	• Avoir traité le cours 2950 "Programmation objet"
Technologies	Visual Studio : C#, création d'un service Windows, tests unitaires, documentation technique



Dans cette mission, toutes les tâches sont obligatoires, en particulier la tâche 4. Cette mission est trop légère pour se permettre d'être tronquée.

Le cahier des charges de l'application Frais GSB stipule que la fiche d'un visiteur est clôturée au dernier jour du mois. Cette clôture sera réalisée par l'application selon l'une des modalités suivantes.

À la première saisie pour le mois N par le visiteur, sa fiche du mois précédent est clôturée si elle ne l'est pas.

Au début de la campagne de validation des fiches par le service comptable, un script est lancé qui clôture toutes les fiches non clôturées du mois qui va être traité.

Nous nous intéresserons ici à la deuxième éventualité.

D'autre part, il est dit que la mise en paiement est faite au 20 du mois suivant la saisie par les visiteurs.

Nous voudrions répondre à ces deux objectifs en développant une application C# avec VS.Net.

Cette application va devoir permettre, au début de la campagne de validation, c'est-à-dire à partir du 1^{er} jour du mois N, la clôture de toutes les fiches créées le mois N-1.

Elle permettra, d'autre part, à partir du 20^e jour du mois N la mise en remboursement des fiches créées le mois N-1.

Une nouvelle application C# doit être créée.

TÂCHE 1 : CRÉATION DE LA CLASSE D'ACCÈS AUX DONNÉES

Dans la nouvelle application, une classe d'accès aux données doit être créée, permettant les fonctionnalités classiques d'accès aux données : connexion à la base (ici ce sera la base MySQL), exécution d'une requête d'administration (insert, update, delete...), gestion d'un curseur (exécution d'une requête type select et gestion du résultat avec passage à la ligne suivante, récupération d'un champ, gestion de la fin du curseur...).
Le but est de créer une classe réutilisable.

TÂCHE 2 : CRÉATION D'UNE CLASSE DE GESTION DE DATES

Cette classe doit être abstraite et ne contenir que des méthodes statiques.

Elle doit contenir au moins les méthodes suivantes :

- *getMoisPrecedent : ne reçoit aucun paramètre et permet de retourner sous forme d'une chaîne de 2 chiffres le numéro du mois précédent par rapport à la date d'aujourd'hui (attention, il faut forcément 2 chiffres : "01" pour janvier, "10" pour octobre...);*
- *getMoisPrecedent : surcharge de la méthode précédente. Cette fois elle reçoit un objet de type DateTime en paramètre et retourne le mois précédent de cette date ;*
- *getMoisSuivant : suivant la même logique que les 2 méthodes précédentes, écrire les 2 méthodes pour le mois suivant ;*
- *entre : reçoit en paramètre deux numéros de jours dans le mois, et retourne vrai si la date actuelle se situe entre ces deux jours ;*
- *entre : surcharge de la méthode précédente. Cette fois un troisième paramètre est reçu de type DateTime et c'est le jour de cette date qui est testée.*

Faites en sorte que le code de la classe soit correctement optimisé (pas de redondance de code).

Réalisez les tests unitaires pour contrôler chaque méthode (cette classe s'y prête de façon idéale).

TÂCHE 3 : CRÉATION DE L'APPLICATION

L'application n'affiche rien. Elle doit juste réaliser le travail demandé sur la base de données, en exploitant les deux classes précédentes, le tout à intervalle régulier (donc en utilisant un timer). Pour les tests, vous utiliserez un intervalle assez court et vous contrôlerez les modifications dans la base de données (pour cela, il faut que vous ayez des informations pertinentes dans la base de données, et que vous trifouilliez les dates pour tester le début du mois, le milieu et la fin.

Rappel des modifications à apporter :

- *récupération des fiches créées du mois N-1 et leur mise à jour, en les mettant à l'état 'CL' ; en supposant que la campagne de validation va se passer entre le 1^{er} et le 10 du mois courant, on va, en comparant les dates, s'assurer que l'on se trouve bien dans cet intervalle-là ;*
- *de la même manière, à partir du 20^e jour du mois, on va mettre à jour les fiches validées du mois précédent en les passant à l'état 'RB' ;*
- *une fois l'application créée (et testée), essayez de voir comment générer la documentation technique sous Visual Studio.*

TÂCHE 4 : CRÉATION D'UN SERVICE WINDOWS

Puisque cette application n'affiche rien et qu'elle doit s'exécuter à intervalle régulier, ce serait une bonne idée qu'elle s'exécute en tâche de fond, sans avoir besoin de la lancer, comme un service Windows. Visual Studio est capable de créer un service Windows (toujours en C#). Cherchez le moyen de le faire et créez ce service.