

TP 3 : Collections et exceptions

Exercice 1

Considérons une classe C# appelée `Personne` ayant les propriétés suivantes :

- Num de type `int`
- Nom de type `String`
- Prenom de type `String`

1. Créez la classe `Personne` avec les trois propriétés.
2. Définissez une méthode `public String GetNomPrenom()` qui retourne le nom concaténé au prénom
3. Dans `Main`, créez quelques instances de la classe `Personne` et ajoutez les à un dictionnaire `Dictionary<String, Personne> personnes = new Dictionary<String, Personne>()` : la clé étant la concaténation des nom et prénom et la valeur étant l'instance de `Personne`.
4. Dans une boucle `for`, affichez la clé pour les éléments de chaque itération d'indice pair et la valeur pour les autres éléments.

Exercice 2

Considérons la classe `Personne` de l'exercice 1.

1. Créez une classe `ListePersonnes` contenant une propriété `Dictionary<String, Personne> Personnes = new Dictionary<String, Personne>()`
2. Écrivez une méthode `public void AjouterPersonne(Personne personne)` qui permet d'ajouter `personne` au dictionnaire `Personnes` : la clé étant la concaténation des nom et prénom et la valeur étant l'instance de `Personne`.
3. Modifiez la méthode précédente pour qu'elle lève une exception de type `PersonneException` (à créer) si les nom et prénom de l'objet `personne` à ajouter existent déjà dans le dictionnaire.
4. Dans `Main` :
 - instanciez un objet de la classe `ListePersonnes`,
 - demandez à l'utilisateur de saisir un numéro de personne positif,
 - redemandez-lui de saisir un numéro tant que la saisie est négative ou nulle,
 - demandez-lui de saisir un nom et un prenom,
 - créez un objet `Personne` avec les valeurs saisies et ajoutez le dans le dictionnaire de `ListePersonnes`,
 - proposez à l'utilisateur de recommencer,
 - si l'utilisateur décide de quitter le programme, affichez tout le contenu du dictionnaire de `ListePersonnes`.

Exercice 3

Considérons une classe `C#` appelée `Nombre` ayant les propriétés suivantes :

- `Var1` de type `int`
- `Var2` de type `int`

1. Créez la classe `Nombre` avec ses deux propriétés

Considérons une deuxième classe appelée `Operation` ayant la propriété suivante :

- `Value` de type `Nombre`

2. Créez la classe `Operation` avec sa propriété.

3. Définissez une première méthode `public int Division()` qui retourne le résultat de la division de `Var1` par `Var2` si ce dernier est différent de zéro. Sinon, elle lève une exception `OperationException` (à créer)

4. Définissez une deuxième méthode `public int RacineDeLaSomme()` qui retourne la racine carrée de la somme de `Var1` et `Var2` si la somme est positive. Sinon, elle lève une exception `OperationException` (à créer)

5. Créez la classe `OperationException`

6. Préparez le ou les constructeurs qui permettront de traiter les exceptions décrites dans les questions précédentes

7. Dans `Main`,

- demandez à l'utilisateur de saisir deux entiers,
- créez un objet de la classe `Nombre`,
- demandez à l'utilisateur de saisir 1 pour avoir le résultat de la division ou 2 pour la racine carrée de la somme,
- utilisez la classe `Operation` pour calculer le résultat de l'opération choisie,
- proposez à l'utilisateur de choisir une autre opération.