



TD 3 Fichiers et Collections

Objectifs :

Utiliser les collections génériques et non génériques.

Utilisation des fichiers

Les 3 premiers exercices sont des applications du cours.

Exercice 1 : Saisir des mots et afficher une phrase V1

Saisir des mots depuis le clavier (un par ligne). S'arrêter à la lecture d'un ".".

Les ranger au fur et à mesure dans un ArrayList

Afficher la phrase ainsi formée

Exercice 2 : Saisir des mots et afficher une phrase V2

Même exercice mais avec la structure imposée : une pile (Stack)

Exercice 3 : Saisir des mots et afficher une phrase V3

Même exercice mais avec la structure imposée : une file (Queue)

Exercice 4 : Utilisation de la Pile : Notation Polonaise inversée

https://fr.wikipedia.org/wiki/Notation_polonaise_inverse

Equation: 3 10 5 + *

Stack			5		
		10	10	15	
Input	3	3	3	3	45
	3	10	5	+	*

Ecrire une calculatrice qui permettrait de faire les opérations élémentaires (*,/,+,-)

`string line0 = "3,3,+,4,*,2,/"` ; qui équivaut à $((3 + 3) * 4) / 2$

Utiliser la collection adéquate.

Exercice 5 : Classe générique : Ecrire une liste chaînée générique

Ecrire une classe générique définie par un attribut de type nœud. Le nœud sera composé d'un élément générique et d'un pointeur sur le nœud suivant (Vu en C au 1^{er} semestre)



Ecrire la classe ListeChainee générique et les méthodes de telle manière que l'on puisse écrire le main suivant :

```
ListeChaine<int> a = new ListeChaine<int>();
a.AddHead(5);
a.AddHead(6);
a.AddQueue(7);
a.Affiche();
a.Insert(4, 1);
a.Affiche();
ListeChaine<string> a1 = new ListeChaine<string>();
a1.AddHead("5");
a1.AddHead("6");
a1.AddQueue("7");
a1.Affiche();
a1.Insert("4", 1);
a1.Affiche();
Console.ReadLine();
```

Exercice 6 : Un annuaire

- créer un fichier (annuaire.txt) avec le bloc note Windows sur le modèle ci-dessous
- préparer un menu comprenant 5 options
 - initialisation de l'annuaire
 - insérer un nouveau téléphone
 - identifier un numéro
 - sauvegarder l'annuaire
 - fin de programme
- rédiger les codes (s'aider par des algorithmes en pseudo-code préalables) pour les 4 premières options
- rédiger l'interface utilisateur sous forme de menu

6-1. Initialisation

- lire le fichier
- insérer les données lues dans une liste triée

6-2. Insérer un nouveau téléphone

- demander le nom et le numéro sur la console
- insérer les données lues dans une liste triée

6-3. Identifier le détenteur d'un numéro

- demander un numéro de téléphone sur la console
- chercher le nom correspondant dans la liste triée
- Afficher le nom correspondant



6-4. Sauvegarder l'annuaire

- sauvegarder chacune des lignes de l'annuaire dans le fichier annuaire.txt

6-5. option : supprimer une entrée

- demander un nom sur la console
- vérifier si le nom est présent dans l'annuaire
- si non présent => afficher un message d'erreur
- si le nom est présent => supprimer l'entrée correspondante et afficher un message de confirmation

Exemple de fichier

0658749621 Kevin

0789674158 Maya

0165754896 Jules

Exercice 7 : List Générique

Une entreprise est définie par son nom, son siège social et l'ensemble de ses salariés.

Chaque personne salariée de l'entreprise est qualifiée par son nom, son prénom, son email, son adresse postale et son salaire. Un numéro de salarié est attribué au moment de l'insertion dans la compagnie. Ce numéro est unique.

Nous souhaitons gérer l'ensemble des salariés de cette entreprise. A savoir, être en mesure de recruter, de licencier, de modifier l'email ou le salaire d'un salarié par exemple.

Ecrire le diagramme de classe UML

Traduire ce diagramme de classe en C#. Utiliser la collection la plus appropriée

Simuler les opérations décrites ci-dessus.