

COURS IFT1166

TRAVAIL PRATIQUE #2

CONSIGNES



1. **Il faut remettre la version électronique. On n'accepte pas des fichiers envoyés par courriel.**
2. **La date de remise est sur Studium.**
3. **Tout travail remis en retard aura une pénalité de 10% par jour de retard.**
4. **Les travaux remis avec 5 jours et plus de retard seront refusés.**
5. **Chaque consigne de cet énoncé qui est non respectée vous fera perdre 5% de la note finale du travail pratique.**

GESTION D'UN CATALOGUE D'ÉPICES ET DES FOURNISSEURS

OBJECTIF

Ce travail pratique consiste à manipuler des **listes doublement chaînées** pour gérer une collection d'épices et leurs fournisseurs associés. Vous devrez lire les données depuis deux fichiers CSV (données séparées par des ;), les stocker dans des structures adaptées, et implémenter une série d'opérations sur ces données. Vous utiliserez des **listes doublement chaînées** pour gérer les épices et les fournisseurs. **Les méthodes de chaque liste devront être testées en premier, avant de procéder aux requêtes.**

STRUCTURE DES FICHIERS

- **Fichier `epices.txt`** : Contient les informations sur les épices (ID, nom, type, prix, quantité, catégorie) et les **IDs des fournisseurs** associés.
- **Fichier `fournisseurs.txt`** : Contient les informations sur les fournisseurs (ID, nom, prénom, courriel, téléphone).

STRUCTURES DE DONNÉES À UTILISER POUR REPRÉSENTER LES DONNÉES DES FICHIERS

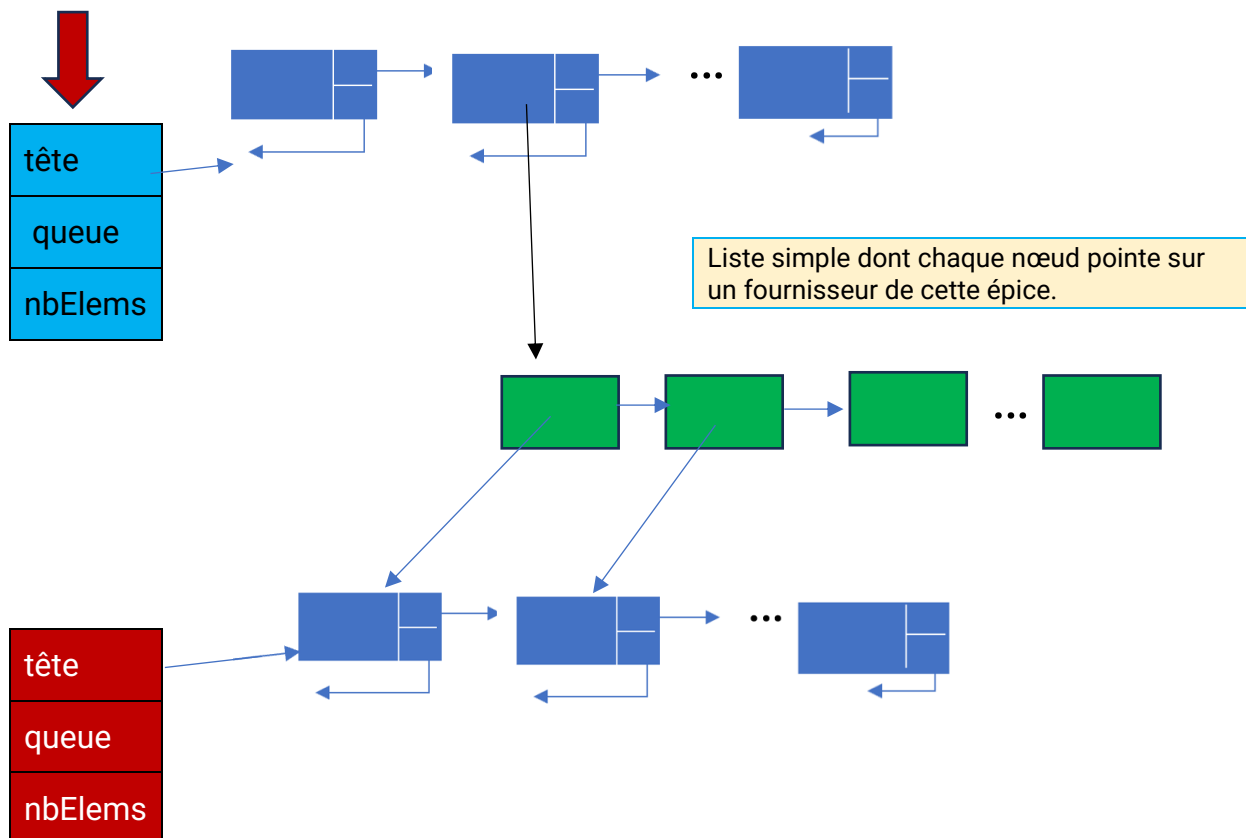
Vous devez avoir les classes ayant les attributs définis préalablement sur les fichiers, un constructeur, un destructeur, les méthodes get et set et une méthode pour retourner les données d'un objet selon le format de votre affichage.

Les classes de données sont **Epice** et **Fournisseur**. Vous avez également besoin d'une classe **DescripteurListeEpices** qui contiendra la tête, la queue et le nombre d'épices dans la liste et une autre classe **DescripteurListeFournisseurs** ayant les mêmes attributs que celui pour les épices.

Tous les attributs des classes sont privés. Ce sont les deux descripteurs qui auront toutes les méthodes pour gérer les listes respectives.

Schéma des structures de données

DescripteurListeEpices **listeEpices** = new **DescripteurListeEpices**(nullptr, nullptr, 0);



DescripteurListeFournisseurs **listeFournisseurs** = new **DescripteurListeFournisseurs**(nullptr, nullptr, 0);

La liste des épices et la liste des fournisseurs sont représentées par une liste doublement chaînée non circulaire.

1. Modélisation des classes (20%)

- Créez une classe **Epice** avec les attributs suivants : id, nom, type, prix, quantite, categorie.
- Créez une classe **Fournisseur** avec les attributs suivants : id, nom, prenom, courriel, telephone.

Évaluation :

- 10% pour la création de la classe Epice et gestion des attributs.
- 10% pour la création de la classe Fournisseur et gestion des attributs.

2. Lecture des fichiers CSV (15%)

- Écrivez des fonctions pour lire les fichiers **epices.txt** et **fournisseurs.txt**. Les données doivent être stockées dans les objets appropriés (Epice et Fournisseur).
- Assurez-vous d'associer les fournisseurs aux épices via leurs **IDs** (présents dans le fichier epices.txt).

Évaluation :

- 7% pour la lecture correcte du fichier epices.txt.
- 8% pour la lecture correcte du fichier fournisseurs.txt et l'association avec les épices.

3. Implémentation des opérations sur la liste doublement chaînée (30%)

Les étudiants devront implémenter et tester les méthodes suivantes sur une **liste doublement chaînée d'épices** avant de procéder aux requêtes.

- **Ajouter une épice.**
- **Supprimer une épice** par ID.
- **Rechercher une épice** par nom ou ID.
- **Afficher toutes les épices** avec leurs informations.

Ensuite, ils devront implémenter et tester ces opérations sur la **liste des fournisseurs** pour chaque épice :

- **Ajouter un fournisseur.**
- **Supprimer un fournisseur** par ID.
- **Afficher tous les fournisseurs** d'une épice donnée.

Évaluation :

- 15% pour les opérations sur la liste des épices.
- 15% pour les opérations sur la liste des fournisseurs.

4. Requêtes sur les données (20%)

Une fois les méthodes testées, les étudiants devront implémenter les requêtes suivantes :

- **Affichage des informations des fournisseurs** (toutes les infos pour une épice donnée (par nom ou par ID).
- **Affichage des informations des épices vendues** par un fournisseur donné (par nom ou par ID).

- **Recherche d'une épice** par catégorie.
- **Affichage des épices triées** par prix (croissant ou décroissant).

Évaluation :

- 5% pour chaque requête correctement implémentée.

5. Défi : Requête complexe (15%)

Implémentez une requête plus complexe comme défi. Vous devez écrire une fonction qui affiche :

- **Les épices vendues par un fournisseur donné, classées par prix**, avec un calcul de la **quantité totale** disponible pour toutes ces épices.

Cette requête combine plusieurs éléments :

- Recherche des épices vendues par le fournisseur.
- Tri des épices selon le prix.
- Calcul de la quantité totale disponible pour ces épices.

Évaluation :

- 10% pour la requête complexe correctement implémentée.
- 5% pour la gestion des cas limites (fournisseur n'existant pas, épices sans fournisseur, etc.).

6. Test du programme avec des exemples (15%)

- Copies écran des tests effectués à placer dans un document **testsTP2** en format **pdf**. Ce fichier doit être placés dans un dossier **documents** créé à la racine de votre projet.
- Testez votre programme avec des données réelles. Utilisez les fichiers fournis (epices.txt et fournisseurs.txt) pour valider que toutes les opérations fonctionnent correctement.
- Testez particulièrement les cas limites : épice ou fournisseur non trouvé, suppression d'une épice sans fournisseur, etc.

Évaluation :

- 10% pour des tests fonctionnels complets.
- 5% pour la gestion des cas limites.

Évaluation finale (100%)

- **Modélisation des classes** : 20%
- **Lecture des fichiers CSV** : 15%
- **Opérations sur la liste doublement chaînée** : 30%
- **Requêtes sur les données** : 20%
- **Requête complexe (défi)** : 15%

Délivrables :

- Le dossier du projet nommé **TP2**