

COURS IFT1166

TRAVAIL PRATIQUE #3

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 DES ÉPICES ET FOURNISSEURS AVEC DES MAPS GÉNÉRIQUES

OBJECTIF DU TRAVAIL

Développer une application en C++ permettant de gérer des **épices** et des **fournisseurs** à l'aide de **maps génériques** basées sur la bibliothèque standard C++ (STL). Le but est de manipuler efficacement les données en respectant les bonnes pratiques de programmation, tout en intégrant des fonctionnalités CRUD et des requêtes spécifiques.

SUJET

Vous devez concevoir une application qui :

1. Lit deux fichiers d'entrée :
 - `epices.txt` : Contient la liste des épices.
 - `fournisseurs.txt` : Contient la liste des fournisseurs.
2. Crée des instances des classes `Epice` et `Fournisseur` à partir des fichiers lus.
3. Stocke ces instances dans deux maps génériques :
 - `map<int, Epice>` pour les épices.
 - `map<int, Fournisseur>` pour les fournisseurs.
4. Propose un menu interactif pour effectuer des opérations CRUD partielles et des requêtes spécifiques.

EXIGENCES TECHNIQUES

- Toutes les sorties numériques doivent avoir **2 décimales suivies du symbole \$**.
- Lors de l'affichage des données dans le terminal :
 - Un **titre** doit être affiché avant les données.
 - Les colonnes doivent être clairement nommées.
 - Les données doivent être alignées sous les colonnes.

FONCTIONNALITÉS DU MENU

1. Lecture des fichiers

- Lire les fichiers `epices.txt` et `fournisseurs.txt`.
- Créer et stocker des instances des classes `Epice` et `Fournisseur` dans leurs maps respectives.

2. Gestion des épices

- Ajouter une nouvelle épice.
- Afficher toutes les épices.
- Supprimer une épice.

3. Gestion des fournisseurs

- Rechercher un fournisseur par son courriel.
- Supprimer un fournisseur.
- Afficher tous les fournisseurs.

4. Requêtes

- Afficher le nombre total d'épices et de fournisseurs.
- Afficher l'épice avec le **prix le plus élevé**.
- Afficher la **moyenne des prix** des épices.

5. Bonus (10%)

- Ajouter dans le code une **structure générique** pour représenter les relations entre épices et fournisseurs :
 - Quel fournisseur fournit quelle épice.
 - Quelle épice est fournie par quel fournisseur.
 - **Note** : Cette déclaration doit apparaître dans votre code avec un commentaire
***** BONUS ***** et mais ne sera pas utilisée dans le programme.

6. Quitter

- Quitter le programme.

1. Classe Générique GestionnaireMap

Encapsule les opérations sur les maps.

```
template <typename Key, typename Value>
class GestionnaireMap {
private:
    map<Key, Value> data;

public:
    void ajouter(const Key& cle, const Value& valeur);
    void supprimer(const Key& cle);
    Value lire(const Key& cle) const;
    void afficherTous() const;
    size_t taille() const;
};
```

2. Classe Epice

Attributs :

- id, nom, type, prix, quantite, categorie.

3. Classe Fournisseur

Attributs :

- id, nom, prenom, email, telephone.

Le menu

```
void afficherMenu() {  
    cout << "\n===== MENU PRINCIPAL =====" << endl;  
  
    cout << "1. Ajouter une nouvelle épice" << endl;  
    cout << "2. Afficher toutes les épices" << endl;  
    cout << "3. Supprimer une épice" << endl;  
    cout << "4. Rechercher un fournisseur par courriel" << endl;  
    cout << "5. Supprimer un fournisseur" << endl;  
    cout << "6. Afficher tous les fournisseurs" << endl;  
    cout << "7. Afficher le nombre total d'épices et de fournisseurs" << endl;  
    cout << "8. Afficher l'épice avec le prix le plus élevé" << endl;  
    cout << "9. Afficher la moyenne des prix des épices" << endl;  
    cout << "10. Quitter" << endl;  
  
    cout << "===== " << endl;  
    cout << "Entrez votre choix : ";  
}
```

EXIGENCES

Bonnes pratiques de programmation

- Code structuré (classes, fichiers .h et .cpp séparés).
- Utilisation appropriée des fonctionnalités de la STL.
- Ajout de commentaires explicatifs pour chaque méthode.
- Respect de la modularité et de la clarté du code.

Gestion des erreurs

- Vérification des fichiers d'entrée (epices.txt, fournisseurs.txt).
- Gestion des exceptions via try-catch pour manipuler les maps et les fichiers.

Rapport de tests

- Produire un fichier PDF contenant :
 - Une capture d'écran pour chaque option testée dans le menu.
 - Une explication des résultats attendus et obtenus.

Partie	Points	Détails
1. Lecture des fichiers	25	- Lecture correcte des fichiers (10). - Création des instances Epice et Fournisseur (10). - Stockage dans les maps génériques (5).
2. Gestion des épices	20	- Ajouter une nouvelle épice (8). - Afficher toutes les épices avec un titre et colonnes bien alignées (8). - Supprimer une épice (4).
3. Gestion des fournisseurs	15	- Rechercher un fournisseur par courriel (6). - Supprimer un fournisseur (5). - Afficher tous les fournisseurs (4).
4. Requêtes	25	- Nombre total d'épices et de fournisseurs (8). - Épice avec le prix le plus élevé (8). - Moyenne des prix des épices (9).
5. Qualité du code	10	- Respect des bonnes pratiques : modularité, clarté et commentaires.
6. Rapport de tests PDF	5	- Présentation claire et captures d'écran des tests.
7. Bonus : Structure générique	10	- Déclaration de la structure générique pour représenter les relations entre épices et fournisseurs, avec le commentaire ***** BONUS *****.

PÉNALITÉS

- Si une option du menu ne fonctionne pas, **-50% des points** pour cette option.
- Le reste des points sera attribué en fonction de l'implémentation correcte des autres éléments.

EXEMPLE DE DONNÉES. LES FICHIERS SONT DANS LE SITE DU COURS.

Fichier epices.txt

1;Cumin;Graines;2.50;100.00;Épices 2;Safran;Poudre;15.00;50.00;Épices
3;Cannelle;Bâtons;4.75;75.00;Épices

Fichier fournisseurs.txt

1;Dupont;Jean;jean.dupont@email.com;1234567890
2;Wang;Li;li.wang@email.com;0987654321 3;Al-Khali;Ahmed;ahmed.khali@email.com;1122334455

Délivrables :

- Le dossier du projet nommé **TP3**. Créez un dossier **documents** dans le dossier **TP3** pour y mettre les copies écran de vos tests. Le fichier de tests sera nommé **TestTP3** et en format **PDF**.