

Documentation Technique :

Gestion de Clients avec

Fichiers Binaires en C#

Table des matières

1. Présentation générale	2
2. Objectifs techniques.....	2
2.1. Concepts abordés.....	2
2.2. Livrables	2
3. Fonctionnalités de l'application.....	2
4. Architecture du projet	3
4.1. Organisation des fichiers	3
4.2. Structure Client.....	4
4.3. Fichier binaire	4
5. Détails des fonctionnalités.....	4
5.1. Ajouter un client.....	4
5.2. Afficher un client	5
5.3. Afficher tous les clients	5
5.4. Supprimer un client.....	5
5.5. Restaurer une fiche supprimée	5
5.6. Compresser le fichier	5
6. Gestion des erreurs.....	5
7. Exemples d'exécution	6
8. Conclusion.....	6

Documentation Technique :

Gestion de Clients avec

Fichiers Binaires en C#

1. Présentation générale

Ce projet consiste à développer une application console en C# pour gérer des fiches clients. Les données sont stockées dans un fichier binaire, permettant un accès rapide et sécurisé. Chaque client est identifié par :

Un identifiant unique (de type entier).

Un nom (chaîne de caractères, enregistré en majuscules).

Un prénom (première lettre en majuscule, reste en minuscule).

Un numéro de téléphone (chaîne de caractères).

L'application offre différentes fonctionnalités accessibles via un menu interactif.

2. Objectifs techniques

2.1. Concepts abordés

Manipulation de fichiers binaires : lecture, écriture, et gestion des données persistantes.

Structures (struct) : organisation des données d'un client sous forme d'entité compacte.

Gestion dynamique : utilisation de listes (List<Client>) pour gérer les clients en mémoire.

Traitement des erreurs : contrôle des saisies utilisateur et gestion des exceptions liées aux fichiers.

Menu dynamique : association des options utilisateur à des méthodes spécifiques.

2.2. Livrables

Code source : structuré et commenté.

Documentation technique : explicative pour le fonctionnement global et les choix techniques.

3. Fonctionnalités de l'application

Voici les fonctionnalités principales accessibles depuis le menu :

Ajouter un client : Permet d'ajouter une nouvelle fiche dans le fichier. Les données saisies sont formatées automatiquement.

Afficher un client : Recherche et affiche un client spécifique à partir de son nom. Gère les doublons et précise la position de chaque client dans le fichier.

Documentation Technique :

Gestion de Clients avec

Fichiers Binaires en C#

Afficher tous les clients : Liste toutes les fiches non supprimées, avec leur position dans le fichier.

Afficher le nombre de clients : Compte et affiche uniquement les fiches actives.

Supprimer un client : Effectue une suppression logique en marquant une fiche comme supprimée (changement du nom en *).

Restaurer une fiche supprimée : Permet de récupérer une fiche marquée comme supprimée.

Afficher les fiches supprimées : Liste les fiches logiquement supprimées.

Compresser le fichier : Supprime physiquement les fiches marquées comme supprimées pour optimiser la taille du fichier.

Quitter l'application : Termine l'exécution de l'application.

4. Architecture du projet

4.1. Organisation des fichiers

Program.cs : Contient le point d'entrée de l'application et le menu interactif.

Client.cs : Définit la structure Client, utilisée pour représenter chaque fiche.

ClientManager.cs : Regroupe les méthodes liées aux fonctionnalités principales (ajouter, modifier, supprimer, etc.).

Documentation Technique :

Gestion de Clients avec Fichiers Binaires en C#

4.2. Structure Client

Le client est défini comme suit :

```
public struct Client
{
    public int Id;
    public string Nom;
    public string Prenom;
    public string Telephone;

    public Client(int id, string nom, string prenom, string telephone)
    {
        Id = id;
        Nom = nom;
        Prenom = prenom;
        Telephone = telephone;
    }
}
```

4.3. Fichier binaire

Le fichier binaire utilisé, nommé clients.dat, contient toutes les fiches clients. Les opérations (lecture, écriture, modification) s'effectuent via des flux binaires (BinaryReader et BinaryWriter).

5. Détails des fonctionnalités

5.1. Ajouter un client

Entrées utilisateur : ID, nom, prénom, téléphone.

Traitement : Le nom est transformé en majuscules, le prénom est mis au format "Première lettre majuscule", et la fiche est ajoutée dans le fichier à la première position disponible (ou en fin de fichier).

Documentation Technique :

Gestion de Clients avec Fichiers Binaires en C#

5.2. Afficher un client

Entrée utilisateur : Nom du client.

Traitement : Recherche toutes les fiches correspondant au nom (insensible à la casse) et affiche leurs informations, y compris leur position dans le fichier.

5.3. Afficher tous les clients

Traitement : Affiche toutes les fiches actives avec leurs informations et leur position dans le fichier.

5.4. Supprimer un client

Entrée utilisateur : ID du client à supprimer.

Traitement : Modifie le champ Nom en * pour marquer la fiche comme supprimée.

5.5. Restaurer une fiche supprimée

Entrée utilisateur : Nom ou ID de la fiche supprimée.

Traitement : Remplace le champ Nom de la fiche par une valeur valide pour la restaurer.

5.6. Compresser le fichier

Traitement : Copie toutes les fiches actives dans un fichier temporaire, puis remplace le fichier d'origine par ce fichier temporaire.

6. Gestion des erreurs

L'application prend en charge plusieurs cas d'erreurs :

Entrées invalides : Vérification et validation des données utilisateur (TryParse).

Fichier inexistant : Création automatique du fichier si nécessaire.

Fiche introuvable : Messages explicites pour guider l'utilisateur.

Documentation Technique :

Gestion de Clients avec Fichiers Binaires en C#

7. Exemples d'exécution

Ajouter un client

Entrez l'ID : 1

Entrez le nom : Dupont

Entrez le prénom : Jean

Entrez le téléphone : 0601020304

=> Client ajouté avec succès.

Afficher un client

Entrez le nom du client à rechercher : DUPONT

Fiche #1 (Position dans le fichier : 0) : ID: 1, Nom: DUPONT, Prénom: Jean, Téléphone: 0601020304

Supprimer un client

Suppression de la fiche ID: 1, Nom: DUPONT, Prénom: Jean, Téléphone: 0601020304

Confirmez-vous la suppression ? (O/N) : O

=> Fiche supprimée logiquement.

Compresser le fichier

Compression en cours...

=> Fichier compressé avec succès.

8. Conclusion

Ce projet est un exercice complet pour manipuler des données binaires et structurer une application en C#. Il permet de travailler sur :

La gestion des fichiers persistants.

L'organisation du code en plusieurs modules.

La robustesse face aux erreurs utilisateur.

Ce programme peut facilement être étendu avec d'autres fonctionnalités, comme l'ajout d'une interface graphique ou l'intégration avec une base de données.

Documentation Technique :

Gestion de Clients avec

Fichiers Binaires en C#

Lien du github :

<https://github.com/bababallz66/MS-Travail-sur-les-fichiers>