

Rapport de conception du projet de gestion de contacts en C++

Introduction

Dans le cadre de notre apprentissage de la programmation orientée objet (POO), nous avons développé une application console pour la gestion de contacts en langage C++. Ce projet avait pour objectif de permettre à l'utilisateur d'effectuer diverses opérations sur des contacts, telles que l'ajout, la modification, la suppression et la recherche, tout en offrant des fonctionnalités avancées comme le tri et la sauvegarde dans un fichier.

Nous avons travaillé en équipe de trois personnes pour concevoir et développer ce projet. Ce rapport détaille la structure du code, le fonctionnement des différentes fonctionnalités, et la contribution de chaque membre de l'équipe.

Objectifs du projet

- **Objectif principal** : Offrir une application de gestion de contacts simple et efficace.
- **Objectifs spécifiques** :
 - Ajouter, modifier, supprimer et rechercher des contacts.
 - Trier les contacts par nom.
 - Sauvegarder et charger les données des contacts via un fichier texte.
 - Proposer une interface console intuitive pour naviguer facilement entre les fonctionnalités.

Structure et fonctionnalités

Structure du projet

1. **Classe Contact** :
 - Représente un contact individuel avec les attributs suivants :
 - nom, telephone, email, adresse.
 - Méthode principale :
 - `afficher()` : Affiche les informations d'un contact.
2. **Liste dynamique de contacts** :
 - Stockée dans un vecteur (`vector<Contact>`) pour une gestion flexible des données.
3. **Fichiers externes** :
 - Sauvegarde des contacts dans un fichier texte (`contacts.txt`).
 - Chargement des données au démarrage de l'application.

Fonctionnalités principales

- **Ajout de contact** : Permet à l'utilisateur de saisir les informations d'un nouveau contact.
- **Modification de contact** : Recherche un contact par son nom et permet de modifier ses informations.
- **Suppression de contact** : Supprime un contact à partir de son nom.
- **Recherche de contact** : Recherche un contact par son nom et affiche ses informations.
- **Tri des contacts** : Trie les contacts par ordre alphabétique.
- **Sauvegarde et chargement** : Enregistre les contacts dans un fichier pour persistance et les charge au démarrage.

Contribution des membres de l'équipe

1. **KONDA Ibrahima- Gestion des données et sauvegarde** :
 - Implémentation des fonctionnalités liées aux fichiers (sauvegarderContacts et chargerContacts).
 - Test de la persistance des données et optimisation du format de sauvegarde.
2. **CONGO Anifatou - Gestion des contacts** :
 - Développement des méthodes d'ajout, de modification, de suppression et de recherche des contacts.
 - Mise en place de la gestion des entrées utilisateur pour garantir des saisies valides.
3. **GNOUMOU S. Jonathan - Interface utilisateur et tri** :
 - Conception du menu interactif et intégration des différentes fonctionnalités.
 - Implémentation de la méthode trierContacts pour le tri alphabétique.
 - Gestion de la boucle principale et des choix utilisateur.

Difficultés rencontrées

- **Gestion des entrées utilisateur** : Il a fallu gérer des saisies incorrectes pour éviter les crashes (noms vides, téléphones non valides).
- **Persistance des données** : L'intégration des fichiers externes a présenté des défis, notamment pour gérer des fichiers absents ou corrompus.
- **Coordination** : Le travail simultané sur des parties interdépendantes (menu principal et méthodes associées) a nécessité plusieurs itérations pour garantir une intégration fluide.

Résultats obtenus

L'application répond aux objectifs fixés :

- Elle permet de gérer une base de contacts de manière intuitive.

- Les données sont sauvegardées et chargées correctement.
- Les fonctionnalités de tri et de recherche offrent une bonne expérience utilisateur.

Conclusion et enseignements

Ce projet nous a permis de mettre en pratique des concepts clés de la programmation orientée objet, tels que :

- L'utilisation des classes et des vecteurs.
- La gestion des fichiers pour la persistance des données.
- La collaboration en équipe pour le développement logiciel.

Chaque membre de l'équipe a apporté une contribution essentielle, rendant possible la réalisation du projet dans les délais impartis.

Annexes

1. Code source complet :

- Classe Contact.
- Méthodes d'ajout, de modification, de suppression, de tri et de recherche.
- Fonctions de sauvegarde et de chargement.

2. Exemple d'utilisation :

- Capture d'écran du menu principal et d'une recherche réussie.

```
Entrez le nom : konda
Entrez le t  l  phone : 54975802
Entrez l'email : kondaibrahima5@gmail.com
Entrez l'adresse : ouagadougou
Contact ajout   avec succ  s !
```

```
---- Menu de Gestion de Contacts ----
```

1. Ajouter un contact
2. Modifier un contact
3. Supprimer un contact
4. Rechercher un contact
5. Trier les contacts
6. Sauvegarder les contacts
7. Quitter

```
Votre choix : 4
```

```
Entrez le nom du contact    rechercher : konda
```

```
Nom : konda
```

```
T  l  phone : 54975802
```

```
Email : kondaibrahima5@gmail.com
```

```
Adresse : ouagadougou
```

```
---- Menu de Gestion de Contacts ----
```

1. Ajouter un contact
2. Modifier un contact
3. Supprimer un contact
4. Rechercher un contact
5. Trier les contacts
6. Sauvegarder les contacts
7. Quitter

```
Votre choix : |
```

