

Mon Projets de Licence D'informatique

DEV1 GitHub

Le projet que vous avez partagé est une application console en Java qui permet de gérer un système d'adhérents avec diverses opérations basées sur l'année et le mois d'adhésion. Voici une explication détaillée du fonctionnement du projet et de ses différentes fonctionnalités :

Structure des données

L'application utilise un tableau bidimensionnel `String[][] adherents` pour stocker les informations sur les adhérents. Chaque ligne de ce tableau représente un adhérent, et chaque adhérent est représenté par trois informations :

- `adherents[i][0]` : L'année de l'adhésion (ex. "2023")
- `adherents[i][1]` : Le mois de l'adhésion (ex. "4" pour avril)
- `adherents[i][2]` : Le nom complet de l'adhérent (ex. "Dany Boon")

Le tableau `adherents` est initialisé avec un certain nombre d'adhérents.

Fonctionnalités principales

1. Afficher tous les adhérents

- Cette option permet d'afficher la liste de tous les adhérents avec leur date d'adhésion (mois/année), qu'ils soient à jour ou non.
- La méthode `afficherTableau()` est utilisée pour parcourir le tableau et afficher chaque adhérent avec ses informations.

```
public static void afficherTableau(String[][] adherents) {  
  
    for (int i = 0; i < adherents.length; i++) {  
        System.out.println("Nom de l'adherent : " + adherents[i][2] + ", Date d'adhesion: " +  
adherents[i][1] + "/" + adherents[i][0]);  
    }  
}
```

2. Afficher les adhérents à jour

- Cette option permet d'afficher les adhérents dont l'adhésion est toujours valide.
- Un adhérent est considéré comme **à jour** si son année d'adhésion est l'année en cours (par exemple 2023) ou si son adhésion est à jour pour décembre de l'année précédente.

- La méthode `afficheradherentsajour()` vérifie si l'année d'adhésion correspond à l'année actuelle ou à décembre de l'année précédente.

```
public static void afficheradherentsajour(String[][] adherents, int anneeEnCours) {
    for (int i = 0; i < adherents.length; i++) {
        if (String.valueOf(anneeEnCours).equals(adherents[i][0])) {
            System.out.println("Nom de l'adherent : " + adherents[i][2] + ", Date d'adhesion: " +
adherents[i][1] + "/" + adherents[i][0]);
        } else if (String.valueOf(anneeEnCours - 1).equals(adherents[i][0]) &&
adherents[i][1].equals("12")) {
            System.out.println("Nom de l'adherent : " + adherents[i][2] + ", Date d'adhesion: " +
adherents[i][1] + "/" + adherents[i][0]);
        }
    }
}
```

3. Afficher les adhérents échus

- Cette option permet d'afficher les adhérents dont l'adhésion est échue. Un adhérent est considéré **échu** si son année d'adhésion est antérieure à l'année en cours et que le mois d'adhésion est avant décembre, ou si son année d'adhésion est inférieure à l'année en cours.
- La méthode `afficherAdherentsEchue()` vérifie ces conditions pour chaque adhérent et affiche les adhérents échus.

```
public static void afficherAdherentsEchue(String[][] adherents, int anneeEnCours) {

    for (int i = 0; i < adherents.length; i++) {
        if ((String.valueOf(anneeEnCours - 1).equals(adherents[i][0]) &&
Integer.valueOf(adherents[i][1]) < 12)) {
            System.out.println("Nom de l'adherent : " + adherents[i][2] + ", Date d'adhesion: " +
adherents[i][1] + "/" + adherents[i][0]);
        } else if (anneeEnCours > Integer.valueOf(adherents[i][0])) {
            System.out.println("Nom de l'adherent : " + adherents[i][2] + ", Date d'adhesion: " +
adherents[i][1] + "/" + adherents[i][0]);
        }
    }
}
```

4. Ajouter ou renouveler un adhérent

- Cette fonctionnalité permet à l'utilisateur d'ajouter un nouvel adhérent ou de renouveler l'adhésion d'un adhérent existant.
- Si un adhérent avec le même nom existe déjà, la date d'adhésion de cet adhérent est mise à jour pour le mois et l'année en cours.
- Si l'adhérent n'existe pas, il est ajouté à la fin du tableau.
- La méthode `ajouteradherents()` permet de gérer cette opération, en prenant en compte le renouvellement ou l'ajout d'un nouvel adhérent.

java

Copier le code

```
public static String[][] ajouteradherents(String[][] adherents, int mois, int anneeEnCours,
Scanner scanner) {
    String nouvelleLigneInput = scanner.nextLine();
    String[] elementsNouvelleLigne = nouvelleLigneInput.split(" ");
    // Vérification de l'entrée et mise à jour ou ajout de l'adhérent
}
```

5. Quitter l'application

- Cette option permet à l'utilisateur de quitter le programme. Lorsque l'utilisateur choisit l'option 5, un message d'adieu est affiché et l'application se termine.

java

Copier le code

```
System.err.println("Au revoir");
Break;
```

Gestion du menu et de l'entrée utilisateur

L'application est interactive et fonctionne via un menu affiché dans la console. L'utilisateur doit choisir une option (1 à 5) et l'application effectue l'action correspondante. Le programme vérifie que l'entrée de l'utilisateur est valide (en utilisant un `while` pour garantir que le choix est compris entre 1 et 5).

Points clés du fonctionnement :

- **Tableau des adhérents** : Utilisation d'un tableau à 2 dimensions pour stocker les informations des adhérents (année, mois et nom).
- **Validation des entrées utilisateur** : Utilisation de la classe `Scanner` pour récupérer l'entrée de l'utilisateur et gestion de l'entrée incorrecte (par exemple, si l'utilisateur entre une date ou un choix invalide).

- **Modification du tableau des adhérents** : Ajout ou mise à jour dynamique du tableau d'adhérents en fonction des actions de l'utilisateur.
- **Affichage des informations** : Les informations des adhérents sont affichées selon l'état de leur adhésion (à jour ou échue).