

PROJET DE PROGRAMMATION JAVA : GESTION D'UN GALA

L'École Supérieure d'Art et de Design souhaite informatiser une partie de la gestion de son gala de fin d'année. Elle vous demande de réaliser une application permettant de gérer les inscriptions et les réservations de tables du dîner.

I. Quelques informations sur l'organisation

Inscription

L'inscription au gala est obligatoire pour avoir accès à la salle ainsi qu'à toute commodité liée au gala. L'application devra permettre aux étudiants de chaque promotion ainsi qu'au personnel de l'école de s'inscrire après identification. Les étudiants devront s'identifier avec leur numéro d'étudiant et les membres du personnel avec un numéro qui leur est attribué à leur arrivée dans l'école.

Réservation des tables

Chaque table du gala possède 8 places. Le comité a décidé de mettre en place trois niveaux de tarifs pour une place à une table de dîner. Le tarif 1 étant le moins élevé est réservé aux étudiants de dernière année, la cinquième année. Le tarif 2 est réservé aux autres étudiants tandis que le tarif 3 sera appliqué au personnel et aux invités.

Personnel

Dix tables sont réservées uniquement pour le personnel. Chaque membre du personnel pourra, au moment de sa réservation, réserver une place supplémentaire pour un invité. Le nombre de places étant suffisant, les réservations seront enregistrées immédiatement. Au moment de la réservation, la personne pourra demander à consulter la composition de toutes les tables réservées au personnel. On affichera alors les noms des membres déjà affectés à chaque table. Si un membre vient accompagné, son nom apparaîtra suivi de la chaîne de caractère « + invité ». La personne pourra alors choisir un numéro de table puis elle précisera le nombre de places qu'elle souhaite réserver (1 ou 2). Elle peut aussi décider de ne pas consulter la composition des tables et choisir directement le nombre de places qu'elle souhaite. Une table lui sera affectée lors de sa réservation. Une fois la réservation effectuée, le montant correspondant aux places sera affiché (on ne s'occupe pas de la gestion du paiement).

Au plus tard 10 jours avant la date du gala, un membre du personnel pourra annuler sa réservation.

Étudiants

Quinze tables sont réservées pour l'ensemble des promotions de l'école. Chaque étudiant pouvant réserver des places pour des invités et le nombre de places au dîner du gala étant limité, le comité a introduit une notion de priorité dans les demandes de réservations. On donnera la priorité aux étudiants de dernière année.

Un étudiant de dernière année pourra réserver jusqu'à 3 places pour 3 invités alors qu'un étudiant d'une année inférieure ne pourra réserver qu'une seule place pour un invité en plus de la sienne.

Les demandes de réservation seront enregistrées suivant leur priorité. Un mois avant le jour du gala, la liste des étudiants dont la réservation aura été acceptée sera générée pour remplir l'ensemble des 120 places. Certains étudiants pourront encore se trouver dans la file d'attente en attendant qu'un étudiant annule sa réservation et libère ainsi une ou plusieurs places.

Les étudiants pourront toujours effectuer des demandes de réservation, ils seront placés dans la file d'attente suivant leur priorité.

Après acceptation des demandes de réservations, les étudiants acceptés devront confirmer leur réservation. Il pourront au moment de cette confirmation demander à consulter la composition de toutes les tables affectées aux étudiants et choisir un numéro de table. Une réservation sera alors enregistrée et l'étudiant sera retiré de la liste des étudiants en attente de confirmation. Le système affichera le montant de la réservation.

Au plus tard 10 jours avant la date du gala, un étudiant pourra annuler sa réservation.

II. Architecture du projet

Les objets de base manipulés dans l'application sont les étudiants, les personnels, les tables ainsi que les réservations. Vous devrez créer les classes correspondantes et en ajouter d'autres, si vous le jugez nécessaire.

Un **étudiant** est caractérisé par un numéro d'étudiant de type `int`, un nom, un prénom, un numéro de téléphone et un mail, chacun étant de type `String`. Chaque étudiant est dans une année de formation (de 1 à 5).

Un **personnel** est caractérisé par un numéro de type `int`, un nom, un prénom, un numéro de téléphone et un mail chacun étant de type `String`.

Une **table** est caractérisée par un numéro de table de type `int` et un nombre de places libres de type `int`. Rappelons que toutes les tables sont des tables 8 places. Nous souhaitons pouvoir enregistrer le nom des participants affectés à chaque table lors des réservations.

Une **réservation** est caractérisée par une date de type `LocalDate`, un numéro de table de type `int`, un nombre de places de type `int` et un montant de type `double`.

On affectera une réservation à l'étudiant ou au personnel qui l'aura créée.

Vous devrez ajouter à chaque classe le ou les constructeur(s) et les méthodes que vous jugerez nécessaires en veillant à identifier les utilisations pouvant entraîner des erreurs. **Vous les documenterez de manière à pouvoir générer la Javadoc.**

Vous jugerez de la nécessité de redéfinir les méthodes héritées de la classe `Object`.

Architecture Modèle Vue Contrôleur :

Créer quatre classes : Gala, Console, Controleur et Main.

La classe Gala

Cette classe centralise la gestion des objets du modèle. Vous devrez choisir les collections de l'API Java adaptées à la gestion des étudiants et des personnels. Il faudra en particulier pouvoir retrouver efficacement un étudiant ou un personnel à partir de son identifiant, enregistrer les étudiants inscrits, les personnels inscrits, les demandes de réservation des étudiants suivant leur priorité, les étudiants dont la demande de réservation aura été acceptée, les tables réservées aux étudiants et celles réservées au personnel. On souhaite pouvoir retrouver une table à partir de son numéro.

Dans la classe Gala, on définira les constantes suivantes :

- les constantes correspondant aux 3 tarifs d'une place : tarif 1 : 10 euros, tarif 2 : 15 euros et tarif 3 : 20 euros.
- les constantes correspondant au nombre de tables affectées aux étudiants (15 tables) et au nombre de tables affectées au personnel (10 tables).

Le constructeur de la classe Gala prendra en paramètre une date de type `LocalDate` correspondant à la date du gala.

La liste de tous les étudiants pouvant s'inscrire au gala est fournie sous forme d'un fichier «`etudiants.csv`». Celle du personnel est fournie sous la forme d'un fichier «`personnel.csv`».

Le constructeur de la classe Gala lit ces fichiers et alimente les objets prévus pour la gestion initiale des étudiants et du personnel.

Il crée également les tables du dîner pour alimenter les collections prévues pour la gestion des tables réservées au personnel (numérotées de 1 à 10) et des tables réservées aux étudiants (numérotées de 11 à 25).

La classe Console

Cette classe gère toutes les interactions avec l'utilisateur. Elle se chargera d'afficher les informations, de récupérer les données saisies par l'utilisateur à l'aide de la classe Scanner et de les transmettre au contrôleur. Le constructeur de la classe prendra donc un contrôleur en paramètre.

La classe Controleur

Cette classe traite les demandes de l'utilisateur, agit sur le modèle ou l'interroge en appelant les services d'un objet de type Gala puis demande à l'objet de type Console de gérer l'affichage et les interactions suivantes.

Le contrôleur connaît donc un objet de type Console et un objet de type Gala.

Le constructeur de la classe Controleur prendra en paramètre une date de type `LocalDate` correspondant à la date du gala. C'est le contrôleur qui va créer l'objet de type Console et créer un objet de type Gala lors du premier lancement de l'application.

Cette classe pourra enregistrer l'identifiant de l'utilisateur connecté pour s'en servir lors des interactions suivantes et en particulier savoir s'il s'agit d'un étudiant ou d'un membre du personnel.

La classe Main

Cette classe contient la méthode `main` qui crée un objet de type `Contrôleur` en passant une date de gala en paramètre.

III. Sauvegarde des données

Nous souhaitons conserver toutes les données du gala de manière à les retrouver à chaque nouveau lancement de l'application. Pour ce faire, lorsqu'un utilisateur quitte l'application, l'objet de type `Gala` contenant toutes les informations devra être sauvegardé par le contrôleur dans un fichier nommé « `gala.ser` » qui sera chargé à chaque lancement de l'application. Le constructeur du contrôleur devra donc créer un objet de type `Gala` lors du premier appel de l'application mais pour les autres appels, il devra charger l'objet `Gala` sauvegardé dans le fichier « `gala.ser` ».

IV. Déroulement des inscriptions et des réservations de tables

Lors du lancement de l'application, afin de vérifier l'état de l'objet `Gala`, vous afficherez le contenu des collections Java qui le composent.

Ensuite, l'utilisateur devra préciser s'il est étudiant ou personnel puis s'identifier avec son numéro. Si l'identification échoue, le système revient au moment où il demande à l'utilisateur s'il est étudiant ou personnel.

Si l'utilisateur n'est pas inscrit, le système lui proposera de s'inscrire ou de quitter l'application. Une fois inscrit, le système affichera le menu donnant accès à la gestion des places du dîner.

Si l'utilisateur est déjà inscrit, le système affichera directement le menu donnant accès à la gestion des places du dîner.

Ce menu proposera 1 – Gérer les places du dîner 2 – Se désinscrire 3 – Quitter

Quand l'utilisateur choisit de gérer les places du dîner, un sous-menu s'affiche. Celui-ci diffère suivant que l'utilisateur est un étudiant ou un membre du personnel.

Cas des personnels

Le sous-menu affiche le nombre de places que l'utilisateur peut réserver et demande s'il souhaite consulter le plan des tables ou pas.

Si l'utilisateur accepte, le plan des tables s'affiche et l'utilisateur doit choisir un numéro de table. Puis il choisit un nombre de places. Si la table ne contient pas assez de places libres, on affiche un message et on revient au sous-menu.

Si l'utilisateur ne consulte pas le plan des tables, il précise le nombre de places qu'il souhaite et une table libre lui sera attribuée.

Si l'utilisateur a déjà fait une réservation, le sous-menu ne s'affiche pas mais le système affiche le nombre de places réservées et le numéro de table. Dans ce cas, il ne peut plus faire de réservation et ne peut pas non plus modifier sa réservation.

Une fois la réservation enregistrée, on quitte l'application.

Cas des étudiants

Les étudiants doivent toujours faire une demande de réservation qui sera placée en attente suivant la priorité définie précédemment. Le sous-menu précisera combien de places l'étudiant peut réserver avant de demander combien de places il souhaite réserver. Si le nombre demandé est inférieur ou égal au nombre de places autorisé, la demande sera enregistrée.

Un mois avant la date du gala, à chaque lancement de l'application, la liste des étudiants dont la demande de réservation aura été acceptée sera mise à jour. Elle dépend du nombre total de places mais aussi des éventuelles désinscriptions d'étudiants.

A partir de ce moment-là, un étudiant ayant fait une demande de réservation et ayant été accepté verra apparaître un autre sous-menu lui demandant de confirmer sa demande de réservation en lui rappelant le nombre de places demandées. Comme pour le personnel, l'étudiant pourra demander à consulter le plan des tables réservées aux étudiants pour choisir son numéro de table. Une fois la confirmation effectuée, on quitte l'application.

Une fois qu'un étudiant a fait une réservation, il ne doit plus avoir accès au sous-menu lui permettant de faire une demande de réservation et ne pourra pas modifier sa réservation.

Informations complémentaires

Création du fichier « *gala.ser* »

Afin de créer le fichier « *gala.ser* » lors du premier lancement de l'application et de ne pas le recréer par la suite, vous devez créer un objet de type `File` et appeler la méthode public boolean `createNewFile()` throws `IOException` qui crée automatiquement un nouveau fichier vide si et seulement un tel fichier n'existe pas déjà.

```
File f = new File("gala.ser");
```

```
f.createNewFile();
```

```
FileInputStream fis = new FileInputStream(f);
```

```
FileOutputStream fos = new FileOutputStream(f);
```

Sérialisation

- Pour sauvegarder un objet de type `Gala`, il faut que la classe `Gala` implémente l'interface `Serializable` et que toutes les classes des objets utilisés dans celle-ci l'implémentent également. En particulier, vous ne pourrez pas utiliser de classe interne anonyme car celle-ci ne pourra pas être déclarée `Serializable`. Dans ce cas, il faut faire appel à une classe interne nommée.
- Afin d'identifier la classe de l'objet sérialisé, il est recommandé d'ajouter à toute classe implémentant l'interface `Serializable` un attribut statique constant de type `long` nommé `serialVersionUID`:

```
private static final long serialVersionUID = 1L;
```