

PROJET DE PROGRAMMATION JAVA : GESTION DE SPECTACLES

A RENDRE LE 08/01/2018

L'objectif est de gérer la programmation et la billetterie de la maison des Arts et de la Culture de la ville d'Arles. Cette maison gère les salles de spectacle. Les spectacles programmés sont de deux types : film et pièce de théâtre.

Un spectacle est identifié par un titre et caractérisé par des interprètes.

Un film est aussi déterminé par son réalisateur et sa durée et une pièce de théâtre par un metteur en scène et un nombre d'entractes.

La maison des Arts et de la Culture est composée de plusieurs salles. Chaque salle porte un nom, a une capacité et un tarif de place fixés. Les salles dans lesquelles sont projetés les films (*salles standard*) n'ont qu'un seul type de place, et ces places peuvent être vendues à prix réduit correspondant à 60% du tarif normal (pour les familles nombreuses, les étudiants, les chômeurs...). Les salles où sont jouées les pièces de théâtre proposent quant à elles deux types de places, la place standard (ou *balcon*) et les fauteuils. Un tarif spécial est attribué aux fauteuils. Aucune réduction ne peut être appliquée dans ces salles.

A un spectacle est associé un ensemble de séances correspondant aux séances planifiées sur une semaine. Une séance est donc caractérisée par un jour de la semaine (entier dans l'intervalle [1. . 7] pour simplifier), un horaire exprimé en heures et minutes, un nombre de places (standard) vendues à tarif normal. Une séance de film est aussi caractérisée par une salle standard et un nombre de places vendues à tarif réduit. Une séance qui correspond à une pièce de théâtre est caractérisée par une salle de théâtre et un nombre de places vendues de type fauteuil. Pour un film, plusieurs séances peuvent être programmées par jour. En revanche, une seule séance par jour est programmée s'il s'agit d'une pièce de théâtre.

1. Créer une classe abstraite `Spectacle` avec les attributs privés `titre` de type chaîne de caractères et `interpretes` de type chaîne de caractères donnant la liste des noms des acteurs ou comédiens.
2. Créer une classe `Film` héritant de la classe `Spectacle` et caractérisée par un attribut `realisateur` de type chaîne de caractères et un attribut `duration` de type `Heure`, la durée sera exprimée en heures et minutes (vous vous inspirerez de la classe `Heure` vue en TD).

3. Créer une classe `PieceTheatre` héritant de la classe `Spectacle` et caractérisée par un attribut `metteurEnScene` de type chaîne de caractères et un nombre d'entractes `nbEntractes` de type entier.

Compléter les classes ci-dessus avec les constructeurs et les méthodes qui vous semblent nécessaires.

4. Créer une classe `Salle` définie par les attributs `nomSalle` de type chaîne de caractères, `capacite` de type `int` et `tarif` de type `double`.
5. Créer une classe `SalleTheatre` héritant de la classe `Salle` avec comme caractéristiques supplémentaires le nombre de fauteuils et le prix du fauteuil. Le `tarif` hérité indique le prix d'une place au balcon (place standard). La somme du nombre de places au balcon et du nombre de fauteuils correspond à la capacité de la salle.
6. Créer une classe abstraite `Seance` avec pour attributs un entier `jour`, un horaire de type `Heure`, et un entier `nbPlacesVenduesTN` indiquant le nombre de places vendues à tarif normal initialisé à 0 au départ. Cette classe fournira les méthodes suivantes:
 - `public abstract int nbPlacesDispo();`
 - `public abstract int totalVendu()` qui retourne le nombre total de places standard vendues.
 - `public abstract double tauxRemplissage()` qui calcule en pourcentage le taux de remplissage de la salle pour la séance.
 - `public void vendrePlacesTN(int nbre)`
7. Créer une classe `SeanceCinema` héritant de la classe `Seance` et ayant comme attributs supplémentaires une salle de type `Salle` et un attribut entier `nbPlacesVenduesTR` indiquant le nombre de places vendues à tarif réduit initialisé à 0 au départ et définissant la méthode suivante :
 - `public void vendrePlacesTR(int nbre)`
8. Créer une classe `SeanceTheatre` héritant de la classe `Seance` et ayant comme attributs supplémentaires `salleTheatre` de type `SalleTheatre` et un entier `nbFauteuilsVendus` et définissant les méthodes suivantes :
 - `public int nbFauteuilsDispo()`
 - `public void vendrePlacesFauteuil(int nbre)`
9. Créer une classe `ProgrammationSemaine` caractérisée par 3 attributs :

- Un attribut de type entier nommé `semaine`.
- Un attribut permettant de d'associer à chaque film l'ensemble des séances de type `SeanceCinema` programmées pendant la semaine pour ce film. Cet ensemble devra pouvoir être ordonné par ordre chronologique (par jour et par heure). Choisir l'implémentation la plus appropriée pour cet ensemble dans la bibliothèque des collections Java (compléter si besoin les classes `Seance` et `Heure`) . Choisir également le type le plus approprié pour l'attribut dans la bibliothèque des collections.
- Un attribut permettant de d'associer à chaque pièce de théâtre l'ensemble des séances de type `SeanceTheatre` programmées pendant la semaine pour cette pièce. Cet ensemble devra pouvoir être ordonné par ordre chronologique (par jour et par heure). Choisir l'implémentation la plus appropriée pour cet ensemble dans la bibliothèque des collections Java. Choisir également le type le plus approprié pour l'attribut dans la bibliothèque des collections.

Cette classe proposera un constructeur permettant de créer une programmation pour une semaine passée en paramètre. Elle proposera également les méthodes permettant les actions suivantes :

- Pour les films : Associer un ensemble de séances de type `SeanceCinema` à un film, récupérer l'ensemble des séances programmées pour un film, récupérer l'ensemble des séances programmées pour un film un jour donné de la semaine, retirer un film de la programmation, récupérer l'ensemble des films programmés pendant la semaine, connaître le nombre de films programmés, vérifier si un film est programmé, supprimer une séance de l'ensemble des séances programmées pour un film, consulter, si elle existe, la séance correspondant à un film, un jour et un horaire.
- Pour les pièces de théâtre : Associer un ensemble de séances de type `SeanceTheatre` à une pièce de théâtre, récupérer l'ensemble des séances programmées pour une pièce de théâtre, récupérer la séance programmée pour une pièce de théâtre un jour donné de la semaine, retirer une pièce de théâtre de la programmation, récupérer l'ensemble des pièces de théâtre programmées pendant la semaine, connaître le nombre de pièces de théâtre programmées, vérifier si une pièce de théâtre est programmée, supprimer une séance de l'ensemble des séances programmées pour une pièce de théâtre.
- Pour chacun des spectacles, on pourra obtenir le taux de remplissage et le chiffre d'affaire.

10. Créer une classe `GestionSpectacles` contenant la méthode `main`.

Deux fichiers au format csv sont fournis. Le fichier `SallesCinema.csv` contient les informations concernant les salles de cinéma et le fichier `SallesTheatre.csv` contient les salles de théâtre.

Dans la méthode `main`, vous commencerez par charger ces fichiers dans deux ensembles : `EnsembleSallesCinema`, l'ensemble des salles de cinéma et `EnsembleSallesTheatre`, l'ensemble des salles de théâtre.

Définir dans la méthode `main` un attribut `lesProgrammations` de type `List<ProgramationSemaine>` qui va permettre de stocker toutes les programmations pour les différentes semaines. L'indice de la programmation dans la liste correspond à la semaine de la programmation.

On proposera ensuite à l'utilisateur de choisir parmi les actions suivantes, et cela en boucle jusqu'à ce qu'il décide de quitter: créer la programmation de la semaine suivante, modifier une programmation existante, vendre des places pour une programmation et enfin consulter les informations concernant les ventes de places comme le chiffre d'affaire et le taux de remplissage.

- Lorsque l'utilisateur choisira de créer la programmation de la semaine suivante, on lui proposera de manière interactive et en boucle de programmer un film ou une pièce de théâtre. La programmation d'un film commencera par la création du film, puis la création d'un ensemble de séances puis enfin l'association du film à cet ensemble de séances. De même pour la programmation d'une pièce de théâtre. La programmation de la semaine ainsi créée sera ajoutée à la liste des programmations des différentes semaines.
- L'utilisateur pourra également choisir de modifier une programmation de semaine existante. On lui demandera quelle semaine il souhaite modifier. Il pourra alors ajouter ou supprimer une séance à un spectacle en fournissant son titre.
- L'utilisateur pourra aussi choisir de vendre des places. Il devra préciser si cela concerne un film ou une pièce de théâtre et pour quelle semaine. Voici un scénario possible dans le cas où il choisit film: l'application affichera l'ensemble des films programmés pour la semaine choisie. Il pourra saisir le titre du film pour lequel il souhaite vendre des places. L'application lui affichera les séances correspondantes. Il pourra alors choisir un jour et un horaire. L'application précisera le nombre de places disponibles pour cette séance. Enfin, l'utilisateur saisira un nombre de places pour lancer la vente.

Vous pouvez ajouter à la classe `GestionSpectacles` les méthodes statiques

qui vous paraissent utiles à une meilleure lisibilité du programme.

Dans toutes les classes créées, vous devrez définir des constructeurs, redéfinir les méthodes `public String toString()`, `public boolean equals(Object obj)` et ajouter des accesseurs et d'autres méthodes que celles indiquées si nécessaire. **Les erreurs pouvant survenir devront être gérées.**

Le projet est à réaliser par binôme. Il est demandé un rapport succinct au format pdf. Ce rapport doit présenter vos choix d'implémentation et vos choix en termes de gestion des erreurs. Vous préciserez également la répartition du travail au sein du groupe.

Les classes devront être commentées.

Déposer les fichiers sources de votre projet sous la forme d'une archive au format `.zip`. Le dépôt se fera dans le cours en ligne en cliquant sur **Dépôt du projet Java**.
Date limite du dépôt : lundi 8 Janvier 2017.