

## PROJET DE PROGRAMMATION JAVA : GESTION DE SPECTACLES

**A RENDRE LE 04/01/2016**

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

Un spectacle est caractérisé par un titre et 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, un entier `capacite`, un réel `tarif` et un entier `nbPlacesStandard`.
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. 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 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 `ProgrammationFilm` permettant de décrire les séances

programmées pour un film et caractérisée par un ensemble de séances de type `SeanceCinema` trié par ordre chronologique (par date de programmation jour, heure). Choisir l'implémentation la plus appropriée pour cet ensemble dans la bibliothèque des collections Java. Compléter si besoin les classes `Seance` et `Heure`. Cette classe doit fournir les méthodes permettant d'ajouter et de supprimer une séance, de rechercher une séance en fonction du jour et de l'heure et de parcourir l'ensemble de séances du film.

Cette classe fournira également les méthodes suivantes:

- `public double tauxRemplissage(int jour, Temps h)` qui calcule le taux de remplissage (en pourcentage) pour la séance repérée par le jour et l'heure donnés en paramètre,
- `public double chiffreAffaire()` qui retourne la recette du film correspondant à l'ensemble des séances.

10. Créer une classe `ProgrammationTheatre` permettant de décrire les séances programmées pour une pièce de théâtre et caractérisée par un ensemble de séances de type `SeanceTheatre`. Cet ensemble sera également trié par ordre chronologique selon la valeur de l'attribut jour.

Cette classe comportera les méthodes suivantes :

- `public double tauxRemplissage(int jour)` qui calcule le taux de remplissage pour le jour passé en paramètre. Ajouter également les méthodes permettant de manipuler l'ensemble des séances.
- `public double chiffreAffaire()` qui retourne la recette du film correspondant à l'ensemble des séances.

Proposer une solution permettant d'optimiser l'implémentation des méthodes de ces deux classes.

11. Créer une classe `ProgrammationSemaine` permettant de décrire les films et les pièces de théâtre programmés sur une semaine et d'associer à chaque film un objet de type `ProgrammationFilm` et à chaque pièce de théâtre un objet de type `ProgrammationTheatre`. Choisir l'implémentation la mieux adaptée dans la bibliothèque des collections Java pour représenter cela. Cette classe sera aussi caractérisée par une semaine de type entier.

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 : Ajouter un film et lui associer une nouvelle programmation. Ajouter des séances à cette programmation, consulter la programmation d'un film, supprimer une séance de la programmation, rechercher un film en connaissant son titre, consulter l'ensemble des

films, consulter les séances correspondant à un film un jour donné, consulter, si elle existe, la séance correspondant à un film, un jour et un horaire.

- Pour les pièces de théâtre : Ajouter une pièce de théâtre et lui associer une nouvelle programmation. Ajouter des séances à cette programmation, consulter la programmation d'une pièce de théâtre, supprimer une séance de la programmation, rechercher une pièce de théâtre en connaissant son titre, consulter la séance correspondant à une pièce de théâtre, un jour donné, si elle existe.
  - Pour chacun des spectacles, on pourra obtenir le taux de remplissage et le chiffre d'affaire.
12. Créer une classe `EnsembleSalles` permettant de stocker l'ensemble des salles standards de la maison des Arts et de la Culture. On pourra ainsi ajouter une salle, rechercher une salle par son nom, parcourir l'ensemble des salles.
  13. Créer une classe `EnsembleSallesTheatre` permettant de stocker l'ensemble des salles de théâtre de la maison des Arts et de la Culture. On pourra ainsi ajouter une salle, rechercher une salle par son nom, parcourir l'ensemble des salles.
  14. Créer une classe `GestionSpectacles` contenant la méthode `main`.

Dans la méthode `main`, on commencera par proposer à l'utilisateur de créer autant de salles standards qu'il le souhaite pour alimenter un objet de type `EnsembleSalle`. De même, on lui demandera de créer les salles de théâtre pour alimenter un objet `EnsembleTheatre`.

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 l'association du film à une programmation de film permettant d'affecter les 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 film ou à une pièce de théâtre.
- 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 `public int hashCode()` et ajouter des accesseurs si nécessaire. Les erreurs pouvant survenir devront être gérées.

*Bonus:* Proposer une solution qui permettrait de stocker dans des fichiers les objets créés comme les salles et la liste des programmations des différentes semaines de manière à pouvoir charger les éléments enregistrés lors d'une session précédente. Cela permettrait, par exemple, de ne pas avoir à recréer les salles à chaque fois qu'on lance l'application mais plutôt d'alimenter un ensemble de salles avec un fichier contenant les objets de type `Salle`. De même pour les salles de théâtre et les programmations des différentes semaines.

Le projet est à réaliser par binôme. Il est demandé un rapport (maximum 4 pages) au format pdf. Ce rapport doit présenter l'ossature du projet ainsi que la répartition du travail au sein du groupe. Les classes devront être commentées.

En plus des fichiers java, vous fournirez également un `jar` de votre application. Le tout sera déposé au format d'archivage (.zip) au début du cours en ligne en cliquant sur **Dépôt du projet Java**. Date limite du dépôt : lundi 4 Janvier 2016.