

# Programmation client-serveur (420-2RP-HY)

## Travail Pratique #1

JavaScript, Classes, UML (5%)

Hiver 2022

### Directives :

1. Ce travail doit être fait en équipe de deux.
2. Remis **par LÉA**.
3. Tous les fichiers demandés doivent être compressés dans un seul document .ZIP
4. Il est de votre responsabilité de conserver une copie de sécurité dans l'éventualité où la lecture des données serait impossible. Cette copie doit être disponible sur demande.

### Critères de correction :

1. Questions bien identifiées.
2. Validité et précisions des réponses aux questions.
3. La lisibilité du code, la documentation et les commentaires explicatifs si nécessaires.
4. Respect des consignes.

### Contexte:

## Le Cinéma

« *Le Cinéma Inc.* » est une entreprise qui gère des réservations de projections de films. Dans ce cours, vous devez développer une application web permettant aux clients de réserver des billets en ligne.

### Tâches pour le TP1:

- Faire le diagramme UML qui présente les classes et ses relations.
- Créer les classes, ses attributs et ses fonctions.
- Remettre un fichier *ÉquipeN.zip* contenant les fichiers suivants : *lesClasses.js*, *lesFonctions.js* et *diagrammeUml.docx*

### Fonctions de vérification des entrées

Dans un fichier *lesFonctions.js*, créer les fonctions suivantes. Celles-ci serviront, plus tard, à vérifier les différentes entrées données par l'utilisateur.

1. *verifierEntier(valeur)*. Cette fonction prend en entrée un string et retourne vraie si les critères sont respectés et faux sinon. Vous devez vérifier que ce string respecte les critères suivants :
  - Il doit représenter un entier.
2. *verifierDate(valeur)*. Cette fonction prend en entrée un string et retourne vraie si les critères sont respectés et faux sinon. Vous devez vérifier que ce string respecte les critères suivants :
  - Il doit être de longueur 10.
  - Il doit représenter une date de format jj-mm-aaaa.
3. *verifierNumeroDeTelephone(valeur)*. Cette fonction prend en entrée un string et retourne vraie si les critères sont respectés et faux sinon. Vous devez vérifier que ce string respecte les critères suivants :
  - Il doit être de longueur 12.
  - Il doit respecter le format CCC-CCC-CCCC
  - Chacun des caractères C doit être un chiffre entre 0 et 9.
4. *verifierCodePostal(codePostal)*. Cette fonction prend en entrée un string et retourne vraie si les critères sont respectés et faux sinon. Vous devez vérifier que ce string respecte les critères suivants :
  - Il doit être de longueur 6.
  - Il doit commencer par une lettre et alterner entre une lettre et un chiffre (LCLCLC).
  - Il doit accepter majuscules et minuscules.

## Classes

Dans un fichier `lesClasses.js`, créer les classes suivantes :

- Artiste : Permet d'instancier les professionnels participants dans la création d'un film.
  - Attributs : nom, prénom, nationalité
- Acteur : Hérite de la classe Artiste. Permet instancier les acteurs d'un film.
  - Attributs supplémentaires : rôle, langue
- Réalisateur : Hérite de la classe Artiste. Permet instancier les réalisateurs d'un film.
  - Attributs supplémentaires : profession, années d'expérience
- Film : Permet d'instancier les films à visionner.
  - Attributs : code, titre, duration, résumé, genre, catégorie, pays, année, date de sortie, producteur, acteurs (liste), réalisateurs (liste), langue, tarif de base
- Salle : Permet d'instancier les salles de projection du cinéma.
  - Attributs : code, nom, type (régulier ou VIP), capacité, facteur tarif.
- Horaire : Permet d'instancier les horaires de projection de films.
  - Attributs : code (H1 à H5), heure début, heure fin, facteur tarif.
- Projection : Permet d'instancier les projections de films.
  - Attributs : salle (objet), film (objet), date, horaire (objet), tarif.
- Réservation : Permet d'instancier les réservations faites par les clients.
  - Attributs : numéro de réservation, projection, code du billet, nom du client, âge, téléphone, adresse courriel, code postal, montant.

Chaque classe doit inclure :

- Un constructeur.
- Une méthode `lireAttribut` pour chacun de ses attributs qui permettront de retourner leurs valeurs.
- Une méthode `modifierAttribut` pour chacun de ses attributs qui permettront de saisir leurs valeurs.

Attributs particuliers :

- La capacité de la salle est par défaut 100
- Le type de la salle peut avoir les valeurs « R » (Régulier, valeur par défaut) et « VIP ».
- La langue du film est par défaut « Français »
- La catégorie du film peut avoir les valeurs « G » (valeur par défaut), « PG », « PG-13 », « R », « NC-17 », « NR ».<sup>1</sup>
- Les facteurs tarif de la salle et de l'horaire ont par défaut la valeur 1
- La valeur du tarif d'une projection doit être calculée en fonction du tarif de base du film et les facteurs tarif de la salle et de l'horaire

Tarif de la projection = tarif de base du film \* (1 + facteur tarif de la salle + facteur tarif de l'horaire)

- L'heure du début d'un horaire peut être 9h, 12h, 15h, 18h ou 21h. L'heure de fin est toujours 3h après l'heure du début.
- Le numéro d'un billet doit être généré en concaténant les codes du film, de la salle et de l'horaire; et un compteur de billet.

No. Billet = code film + code salle + code horaire + compteur

## Diagramme UML

Représenter les classes et leurs relations dans un Diagramme de classes UML. Insérer l'image dans un fichier Word.

## Tests unitaires

Afin de vous assurer que vos classes et vos fonctions ne comportent pas d'erreur, vous devez écrire le code de tests.

1. Dans le fichier `lesFonctions.js`, faites **2 tests par fonction** afin de vous assurer qu'elles retournent le bon résultat. Pour ce faire, choisissez une entrée à votre fonction. L'exécution de ce code doit afficher dans le terminal:
  - L'entrée
  - Ce que devrait retourner la fonction
  - Ce que la fonction retourne réellement

Par exemple, si j'exécute le code `verifierNumeroDeTelephone("222-333-4444")`, je dois voir apparaître :

Entrée : "222-333-4444"

Résultat attendu : Vrai

Résultat obtenu : Vrai

2. Dans le fichier `lesClasses.js`, ajouter du code pour instancier toutes les classes.

---

<sup>1</sup> <https://cineplex.zendesk.com/hc/fr/articles/115005901788-Que-signifient-les-classifications-de-film->

**Barème d'évaluation :**

Fonctions de vérification des entrées .....	5%
Classes .....	15%
Diagramme UML .....	10%
Tests, documentation et commentaires .....	10%
<b>Total *</b> .....	<b>40%</b>

\*Sera ramené à 5%

PDEA #2: (L'évaluation du français) Lors d'activités d'évaluation sommative en classe ou hors classe (programmes, incluant les commentaires), documentation, rapport de laboratoire, rapport de stage, examen), une pénalité maximale de 10 % peut être retranchée de la note finale de ladite évaluation (le barème étant de 0,5%/erreur).

PDEA #4 : **Toute évaluation sommative remise après la date d'échéance fixée se voit attribuer la note zéro pour les étudiants de 2e à 6e session.**

Note : **Tout travail plagié en partie ou en totalité se verra attribuer une note totale de 0 %.**