

- Barème donné à titre indicatif
- Le respect des conventions de nommage et des principes de POO vus en cours sera pris en compte dans la notation.

Dans Eclipse, créez un projet à partir de l'archive `CC_2020_Mardi.zip` : `Import` → `General` → `Existing Projects into Workspace`, puis cochez la case `Select archive file`, et choisissez le fichier correspondant.

Cette archive contient des packages et des classes dans lesquels vous pouvez taper vos réponses aux exercices, ainsi que du code que vous pouvez utiliser directement sans avoir à le modifier. Par exemple, dans la classe `exo1.UtilTab`, il y a la méthode :

```
public boolean minMax(int n, int[] tab1, int[] tab2) {  
    throw new UnsupportedOperationException();  
}
```

Remplacez le code `throw new UnsupportedOperationException();` par votre réponse à la question 1. de l'**Exercice I**.

Pensez à renommer le projet en y ajoutant votre nom, afin de faciliter l'identification des « copies ». Pour cela, clic droit sur le nom du projet, `Refactor` → `Rename`. À la fin de l'épreuve, exportez votre projet sous forme d'archive `zip` afin de l'uploader sur Moodle : `Export` → `General` → `Archive File`.

Exercice I (4 points)

Dans la classe `exo1.UtilTab`,

- (1) 1. écrivez une méthode qui prend en entrée un nombre entier et deux tableaux d'entiers, et détermine si le nombre donné est plus petit que le plus petit élément du premier tableau, et plus grand que le plus grand élément du second tableau.
- (1) 2. écrivez une méthode qui prend en entrée un nombre entier et un tableau d'entiers, et détermine si le nombre donné est la médiane des éléments du tableau.
- (2) 3. écrivez une méthode qui prend en entrée deux tableaux d'entiers de même longueur n , et retourne un nouveau tableau d'entiers de longueur $2 \times n$, tel que les éléments d'indice pair sont ceux du premier tableau, et ceux d'indice impair sont ceux du second tableau.

Par exemple, avec `tab1` le tableau `[1,2,3]` et `tab2` le tableau `[4,5,6]`, la méthode retourne le tableau `[1,4,2,5,3,6]`.

On ne prendra pas la peine ici de vérifier si les deux tableaux sont bien de la même longueur.

Exercice II (8 points)

Un cinéma souhaite un logiciel qui permet de gérer les ventes de places. Pour cela, dans le package `exo2`, répondez aux questions suivantes.

- (1) 1. Créez une classe `Salle` qui permet de représenter une salle du cinéma, avec les informations suivantes :
 - son numéro,
 - le nombre maximal de spectateurs qu'elle peut contenir,
 - est-ce qu'elle diffuse des films en 3D ou pas.
- (2) 2. Créez une classe `Seance`, qui permet de représenter la diffusion d'un film précis, dans une salle précise, à un horaire précis. Pour simplifier les choses, on considère qu'une séance commence toujours à l'heure pile (par exemple, 14h00 ou 15h00, mais pas 14h15 ni 14h30), et dure précisément deux heures. Cette classe doit contenir une méthode qui permet d'indiquer qu'une place a été vendue. Si la salle est déjà pleine, cette méthode retourne **false**, et bloque la vente de la place.
- (5) 3. Créez une classe `Cinema`, qui permet de représenter les différentes informations nécessaires au fonctionnement du cinéma :
 - les salles du cinema,
 - la liste des séances.

Lors de la création d'une séance, le programme doit vérifier qu'il n'y a pas un chevauchement avec une autre séance prévue dans la même salle. Si c'est le cas, la séance n'est pas créée.

La classe `Cinema` doit permettre de vendre une place pour une séance particulière, et permet d'afficher également :

- le taux de remplissage moyen des séances (pour chaque séance, le taux de remplissage est le nombre de places vendues divisé par le nombre de places disponibles dans la salle),
- le chiffre d'affaire sur l'ensemble des séances, en sachant qu'une place coûte 10 euros au tarif normal, et 12 euros pour une séance en 3D.

Votre code doit fonctionner avec la méthode `main` de la classe `exo2.TestCinema`.

Exercice III (4 points)

Dans le package `exo3`, créez un programme qui permet d'interagir au clavier avec l'utilisateur, afin de lui proposer les fonctionnalités suivantes :

- demander à l'utilisateur de taper un mot, et lui donner en réponse le mot « miroir » ;
- demander à l'utilisateur de taper un mot, et lui indiquer si le mot en question est un palindrome ;

- demander à l'utilisateur un nombre entier n , puis de taper n mots, et lui donner la liste des mots triés dans l'ordre alphabétique.

Vous êtes invités à utiliser les méthodes fournies dans `exo3.UtilMots`, votre tâche consiste donc à gérer l'interface textuelle pour communiquer avec l'utilisateur.

Exercice IV (4 points)

Sans utiliser l'interface `java.util.List` et les classes qui l'implémentent, définissez dans le package `exo4` une classe qui représente une liste d'entiers positifs ou nuls. Cette classe doit (au moins) contenir des méthodes pour

- ajouter un élément dans la liste,
- vérifier si un nombre est présent dans la liste,
- obtenir la position d'un nombre dans la liste,
- obtenir le nombre présent à une position donnée.

Votre code doit fonctionner avec la fonction `main` présente dans `exo4.Test`.