

# Examen de rattrapage Java

---

## Exercice 1 : Générateur de code HTML

La figure suivante représente la structure minimale d'un code HTML :

```
<html>
<head>
<title> TITRE </title>
</head>
<body>
<!-- Tout le contenu de la page -->
</body>
</html>
```

Cette structure est composée de :

1. l'élément principale `<html>`;
2. deux éléments obligatoires : l'entête `<head>` et le corps `<body>`;
3. l'élément `<title>` dans l'entête qui précise le titre de la page HTML; et
4. Les éléments du corps représentant le contenu de la page.

Vous voudrions créer en Java un générateur de code HTML :

1. Écrire la méthode `genererTitre` qui prend en paramètre le titre de la page et retourne le code HTML de l'élément `title`.
2. Définir la méthode `genererEntete` qui retourne le code HTML du contenu de l'entête. Cette méthode doit prendre en paramètre le code générée par `genererTitre`
3. Définir le méthode `genererContenu` qui génère le code HTML de trois types de contenu au choix : `<h1>`, `<h2>` ou `<h3>`. Le type et le contenu des éléments doivent être fixés en paramètre.

Exemple d'élément h :

```
<h1>Contenu de l'element type 1</h1>
<h2>Contenu de l'element type 2</h2>
<h3>Contenu de l'element type 3</h3>
```

4. Écrire la méthode `genererCorps` qui retourne le contenu du corps HTML. Elle doit prendre en paramètre le code généré par `genererContenu`
5. Dans la méthode principale, écrire un programme qui affiche le code suivant, en utilisant les méthodes définies précédemment :

```
<html>
<head>
<title> Master 2 ESE </title>
</head>
<body>
<h1>Ceci est l'examen de rattrapage</h1>
</body>
</html>
```

## Exercice 2 : Répertoire téléphonique

Soit l'énoncé suivant :

- Un répertoire téléphonique qui regroupe plusieurs contacts;
  - Chaque contact est une personne qui est définie par son nom, prénom et pseudonyme.
-

- Chaque contact doit avoir, en plus des informations personnelles, plusieurs numéros téléphoniques et plusieurs adresses électroniques.
  - Un numéro de téléphone doit être constitué de 10 chiffres, le premier chiffre étant obligatoirement 0.
  - L'adresse électronique ne doit pas commencer par un chiffre, ne doit pas contenir les caractères : \* / + - et doit contenir le caractère @.
1. Définir les classes **Personne**, **Contact**, **Numero** et **EEmail**, ainsi que les setters, getters et les constructeurs de chaque attribut, tenant compte de l'énoncé.
  2. Dans la classe **Repertoire**, définir la méthode principale **main** () qui crée un contact de votre choix.

## Exercice 3 : File d'attente

Soit l'interface suivante :

---

```
public interface File {  
    public int size(); //Retourne le nombre d'element  
    public boolean isEmpty(); //Verifier si la file est vide  
    //Ajouter un element a la fin de la file  
    public void enfiler(Object o);  
    //Supprime et retourne le premier element de la file.  
    public Object defiler();  
}
```

---

1. Définir une classe qui implémente l'interface **File**. Cette structure modélise un file d'attente dans laquelle plusieurs données sont stockées (utiliser les tableaux dynamiques), l'ajout des données se fait en fin de liste, et la suppression se fait sur le premier élément de la liste.
2. Définir une méthode qui affiche tous les éléments de la file.