

## EVALUATION

Evaluation de fin du module sur les techniques de programmation. L'évaluation se décompose en deux étapes. Une partie où vous développez un programme qui lit des questions et fait répondre un utilisateur et sauvegarde le résultat dans un fichier html, une autre partie où vous répondez correctement aux questions.

Nom du projet : Evaluation\_M1

Package : fr.beweb.evaluation.m1.ville

Les documents fournis sont sous la forme d'une archive à décompresser dans le dossier de votre choix.

Vous disposez d'un dossier libraries dans lequel se trouve le fichier EvaluationLib.jar que vous devez importer dans votre projet en cliquant droit sur « Librairies » et « add jar file ». Dans ce même dossier se trouve le dossier javadoc où vous avez la documentation de la bibliothèque.

A la racine du dossier vous trouverez le fichier de questionnaire au format texte, un fichier css et un fichier jpg.

Le fichier de sortie sera créé lors l'exécution du programme. Pensez à choisir le bon chemin du fichier de sortie.

Vous avez accès à toutes les ressources disponibles, slack, internet, supports de cours et exercices.

### **1. PARTIE ETUDE – PREPARATION (15MIN)**

---

Vous vous assurez d'avoir lu tous les fichiers, et d'avoir compris de quoi il en retourne. Si vous avez des questions c'est à ce moment qu'il faut les poser au formateur.

## 2. PARTIE TECHNIQUE – ALGORITHMES (2h30)

---

En vous appuyant sur la bibliothèque fournie, vous devez implémenter un générateur d'évaluation en console. Vous avez à votre disposition plusieurs fichiers vous permettant de réaliser cette application.

- Le fichier « questionnaire.txt » : Dispose d'une série de questions classées par thématiques. Il existe deux types de questions, les Q/R et les QCM. Les choix des QCM seront toujours de 4. Vous devez étudier la structure du fichier avant de commencer à coder.
- Le fichier « style.css » : Vous y trouverez le style appliqué au rendu final afin que vous en déduisiez la structure du rendu.
- Le fichier « snapEval.jpg » : Screenshot du rendu final. Avec le css et le questionnaire vous arriverez à choisir les bonnes balises pour la génération du fichier html.

La bibliothèque fournie vous propose tous les outils pour mener à bien l'évaluation dans le temps imparti :

```
// vous stockez la bibliothèque dans la variable eval  
CoreEvaluation eval = EvaluationSystem.getEvaluation();
```

Vous trouverez la liste des méthodes disponibles dans l'interface « CoreEvaluation ». Vous pouvez aussi (et c'est la meilleure approche) taper : « eval. » si NetBeans ne vous propose pas la liste des méthodes disponibles, il vous suffit de faire ctrl+espace.

Vous disposez aussi d'une classe de type Tag correspondant à une balise. Cet objet possède une liste de méthodes utiles pour définir la structure de votre rendu final.

Exemple de code :

```
// création des balises
Tag root = new Tag("html");
Tag head = new Tag("head");
Tag body = new Tag("body");
// style doit être dans le même dossier que index.html
head.addStylesheet("style.css");
// insertion de la balise head dans la balise html
head.insertInto(root);
// insertion de la balise body dans la balise html
body.insertInto(root);
// insertion d'une div dans la balise body
Tag a = new Tag("div");

a.insertInto(body);
```

Cette classe possède une méthode `getHtml()` à invoquer sur le tag `html`, elle retourne le code html en chaîne de caractère à écrire dans le fichier `index.html`.

```
// insere le code html dans la variable codeHTML

String codeHTML = root.getHtml();
```

L'objectif est de rentrer les questions et les réponses dans le fichier `html` et faire correspondre le rendu avec le screenshot fourni.

Vous devez être en mesure de mettre en place les bonnes balises aux bons endroits.

### 3. PARTIE THEORIQUE – QUESTIONNAIRE (15MIN)

---

L'application lit le fichier questionnaire et propose de répondre aux questions, les questions et les réponses sont organisées grâce à la structure des balises.

Lorsque le programme est fonctionnel, vous répondez correctement aux questions et envoyez une copie de votre fichier `html` à votre formateur avec votre adresse ip et l'url de votre ressource.

Vous ferez en sorte que votre fichier soit disponible sur votre serveur prêt à répondre à des requêtes du réseau local.

### 4. TIPS

---

Afin de vous aider dans votre tâche voici quelques tuyaux :

- Mettez la liste des questions dans une liste de « string » ce sera plus simple pour travailler :

```
// initialisation d'une liste de String
List<String> questions = new ArrayList<>() ;
// ajoute une chaine dans la liste

questions.add("ma chaine");
```

- Un objet String a une méthode permettant de savoir si un caractère existe à un index dans la chaine :

```
// initialisation d'une variable de type String
String message = "Hello world !!!!";

char c = message.charAt(0); // la variable c vaut le caractère 'H'

boolean b = message.charAt(2) == 'l'; // le caractère 'l' est bien à l'index
2 donc b est vrai !
```

- L'objet que vous allez utiliser, stocké dans la variable « eval », contient la méthode « interact() » pour la gestion des questions / réponses. Elle vous évite de vous embêter avec l'objet Scanner, donnez-lui en argument la question elle retourne la réponse de l'utilisateur :

```
// votre objet de base pour l'évaluation
CoreEvaluation eval = EvaluationSystem.getEvaluation();

// la méthode interact retourne la réponse à la question passée en argument
String r = eval.interact("ma question");
// Regardez l'objet eval, cette méthode est surchargée
```

- Il y a plusieurs autres méthodes fournis par les objets Tag et CoreEvaluation, pour effectuer votre travail, Regardez les avant de travailler.
- N'hésitez pas à déclarer pleins de variables si cela vous semble astucieux.
- Déroulez votre algo, sur papier ou en commentaire avant de commencer.