

# OPENCLASSROOMS

## Mettez votre logique à l'épreuve

140 heures

### Énoncé

#### Pré-requis

Ce projet ne nécessite pas de pré-requis autres que ceux d'entrée dans le parcours de formation de Développeur d'Application Java JEE.

#### Contexte

Vous voilà prêt à entrer de le vif du sujet : programmer !

Dans ce premier projet, je vous propose de créer une application proposant des jeux de logique. Vous aurez à développer plusieurs variantes autour de la recherche de combinaisons secrètes :

- La recherche d'une combinaison à l'aide d'indications +/-
- Le célèbre Mastermind. (Si vous voulez, vous pouvez vous baser sur des chiffres au lieu des couleurs.)

Plus d'informations sur <http://fr.wikipedia.org/wiki/Mastermind>

#### Recherche +/-

Le but : découvrir la combinaison à x chiffres de l'adversaire (le défenseur). Pour ce faire, l'attaquant fait une proposition. Le défenseur indique pour chaque chiffre de la combinaison proposée si le chiffre de sa combinaison est plus grand (+), plus petit (-) ou si c'est le bon chiffre (=).

L'attaquant doit deviner la combinaison secrète en un nombre limité d'essais.

```
(Combinaison secrète : 1234)
```

```
Proposition : 4278 -> Réponse : ---
```

```
Proposition : 2214 -> Réponse : -+=
```

```
...
```

# OPENCLASSROOMS

## Mastermind

Le but : découvrir la combinaison à x chiffres/couleurs de l'adversaire (le défenseur). Pour ce faire, l'attaquant fait une proposition. Le défenseur indique pour chaque proposition le nombre de chiffre/couleur de la proposition qui apparaissent à la bonne place et à la mauvaise place dans la combinaison secrète.

L'attaquant doit deviner la combinaison secrète en un nombre limité d'essais.

```
(Combinaison secrète : 1234)
```

```
Proposition : 4278 -> Réponse : 1 présent, 1 bien placé
```

```
Proposition : 6274 -> Réponse : 2 bien placés
```

```
...
```

## Travail demandé

Vous devez développer une application en Java, soit en mode console soit avec Swing, permettant de lancer les différents jeux.

Chaque jeu pourra être joué selon 3 modes :

1. Mode *challenger* où vous devez trouver la combinaison secrète de l'ordinateur
2. Mode *défenseur* où c'est à l'ordinateur de trouver votre combinaison secrète
3. Mode *duel* où l'ordinateur et vous jouez tour à tour, le premier à trouver la combinaison secrète de l'autre a gagné

Voici le fonctionnement normal attendu dans l'application :

1. Au démarrage, l'utilisateur doit choisir le jeu auquel il veut jouer parmi les choix proposés.
2. L'utilisateur sélectionne le jeu et le mode de son choix. L'application lance le jeu sélectionné.
3. L'utilisateur joue. S'il perd, l'application affiche la solution.
4. À la fin de la partie, l'utilisateur peut choisir :
  1. de rejouer au même jeu
  2. de lancer un autre jeu (retour à l'écran de choix des jeux du début)
  3. de quitter l'application

# OPENCLASSROOMS

Il doit être possible de lancer l'application dans un mode "développeur". Dans ce mode la solution est affichée dès le début. Cela permet de tester le bon comportement de l'application en cas de bonne ou de mauvaise réponse de l'utilisateur. Ceci est à réaliser avec les mécanismes suivants :

- Passage d'un paramètre au lancement de l'application
- Propriété spécifique dans le fichier de configuration

Un fichier de configuration (`config.properties`) permettra de paramétrer l'application, notamment :

- Pour chaque jeu :
  - le nombre de cases de la combinaison secrète
  - le nombre d'essais possibles
- Pour le Mastermind :
  - le nombre couleur/chiffre utilisables (de 4 à 10)

Un fichier de configuration (`log4j.xml`) permettra de paramétrer les logs de l'application. La gestion des logs se fera avec *Apache Log4j*.

## Livrables attendus

Vous livrerez, sur [GitHub](#) ou [GitLab](#) (dans un seul dépôt Git) :

- le code source de l'application
- le fichier de configuration
- une documentation succincte (un fichier `README.md` suffit) expliquant comment compiler et lancer l'application

Pour ce projet, la validation s'effectuera par un mentor validateur lors d'une soutenance filmée.

Votre mentor accompagnateur fera la demande de soutenance quand vous aurez de la visibilité sur la complétion du projet.

Soignez particulièrement le code source de votre application : indentation, formatage, nom des variables, commentaires...

Le projet est validé tant sur le fond (application fonctionnelle) que sur la forme (organisation technique, présentation du code...).

# OPENCLASSROOMS

Vous donnerez un accès en lecture à votre dépôt Git à votre mentor et au mentor qui vous fera passer la soutenance

## Soutenance

### Avant la soutenance

Quelques jours avant la date de la soutenance (en général 3-4 jours avant), vous transmettez tous vos livrables au mentor qui vous fera passer celle-ci, afin qu'il puisse en prendre connaissance en amont et éventuellement tester votre application.

### Déroulement de la soutenance

La soutenance, d'une durée de 20-25 minutes, se déroulera en deux parties :

- Partie 1 – 15 minutes : **Présentation de l'application**
  - [~ 5 minutes] Vous ferez une démonstration rapide de votre application
  - [~ 10 minutes] Vous présenterez votre solution technique et les points particuliers de votre développement (à vous d'être pertinent sur les points à aborder)
- Partie 2 – 5-10 minutes : **Retour sur la soutenance**
  - Le mentor pourra vous demander d'approfondir certains aspects ou vous questionner sur vos livrables
  - Le mentor vous fera un debrief sur votre prestation en soutenance

## Compétences à valider

- Mettre en oeuvre les concepts fondamentaux de la programmation en Java