

Durée du CC : 1h30.

La Fiche-Triche JavaScript est le seul document autorisé.

Les deux parties sont indépendantes ; on pourra les traiter dans l'ordre de son choix. Pour le rendu, deux fichiers sont attendus : `partie1.js` pour la première partie, et `partie2.js` pour la seconde.

1 Apiculture

Le code pour cette section est à rendre dans un fichier nommé `partie1.js`.

Question 1 Écrire une classe `Ruche` possédant les deux attributs suivants :

- `abeilles`, qui représente le nombre d'abeilles dans la ruche, et
- `miel`, qui indiquera la quantité de miel, en grammes, que contient celle-ci.

On considérera qu'une ruche ne contient aucun (autrement dit, 0 grammes) miel à sa création. Le constructeur de cette classe aura donc un unique argument : le nombre d'abeilles.

Question 2 Implémenter la méthode `production()` de `Ruche`. Cette méthode n'attend aucun argument, et au bout de 3 secondes (sans bloquer) augmente la quantité de miel de 5 grammes pour chaque abeille présente dans la ruche.

Question 3 Écrire une classe `Apicole`, qui représentera une entreprise apicole (c'est-à-dire, une entreprise productrice de miel). Cette classe possédera deux attributs : un `nom` et un tableau `ruches` de ruches.

Écrire un constructeur complet (à deux arguments) pour cette classe.

Question 4 Implémenter la méthode `production()` de la classe `Apicole`, qui appellera la méthode `production()` de chaque ruche de `ruches`.

On utilisera une boucle *for* pour répondre à cette question.

Les abeilles ayant besoin de miel pour vivre, les apiculteurs et apicultrices ne récoltent jamais l'intégralité du miel présent dans une ruche. Pour les besoins de la modélisation, on considérera qu'à chaque récolte, on laisse, si possible, 100g de miel dans chaque ruche. En particulier, si une ruche possède moins de 100g de miel, alors aucun miel n'y sera récolté.

Question 5 Implémenter la méthode `recolte()` de la classe `Apicole`. Un appel à cette méthode devra programmer la récolte (c'est-à-dire, mettre à jour la quantité de miel) de chaque ruche, dans l'ordre, avec un délai de 5 secondes entre chaque récolte : il s'agit du temps nécessaire à l'apiculteur pour faire sa récolte.

Pour répondre à cette question, on n'utilisera ni boucle *for*, ni boucle *while*. On pourra, si besoin, ajouter une méthode à la classe `Ruche`.

2 King's Landing

Téléchargez le fichier `King's_Landing.html` sur Eprel.

Le code de cette section sera à rendre dans le fichier `partie2.js`. Le fichier `King's_Landing.html` n'est pas à rendre.

Question 6 Ajouter une balise au fichier `King's_Landing.html` permettant d'exécuter le code contenu dans le fichier `partie2.js`.

Pour les questions suivantes, on ne touchera plus au fichier html : les balises devront être créées ou modifiées dynamiquement depuis le script.

Question 7 Créer des balises `<p>` contenant le texte “Nouvelle section.”, et les placer directement après chaque `<h2>` de la page.

Question 8 Faire en sorte que lors d'un clic droit sur un lien `<a>`, ce lien soit désactivé : on veut alors qu'il soit remplacé par une balise `<p>` contenant le même texte que le lien d'origine. On utilisera une lambda-expression pour répondre à cette question.