

Complément sur les abeilles : Simulation de la vie d'une ruche

Si vous avez fini le TP *Découverte de l'héritage & du polymorphisme avec les abeilles - Introduction à la qualité de code*, vous pouvez continuer à travailler sur des problèmes de conception et programmation objet à l'aide de ce mini-projet. Si vous ne finissez pas ce projet aujourd'hui, gardez-le sous la main comme exercice supplémentaire pour les TP que vous finirez plus tôt à l'avenir 😊
Il est temps de donner un peu d'interactivité à vos programmes.
(en mode console, rassurez-vous la ressource **JavaFX** ne devrait plus tarder à commencer pour les IHM en mode graphique 😊)



Ce premier exercice doit se faire en **mode déconnecté**



De l'abeille à la ruche ... : Analyse & Conception



Votre voisin, qui souhaite devenir apiculteur à ses heures perdues, vous demande de lui développer un projet lui permettant de simuler d'une manière très simple la vie d'une ruche, avant de se lancer dans l'achat de sa première ruche, une ruche identique à celle représentée sur l'image ci-contre (voir annexe pour plus de détail sur les composants de cette ruche).

Dans cette simulation, il souhaite pouvoir donner un nom à sa ruche, se souvenir de l'année de mise en service de sa ruche et absolument connaître le nombre de hausses à miel dont la ruche dispose actuellement. Le nombre de hausses a une influence sur la production de miel.

Il vous indique qu'il souhaite pouvoir ajouter des hausses à sa ruche.

Pour l'instant, ce sont les seules informations qui l'intéressent concernant sa ruche, il vous fera part de ses autres besoins au fil du temps... 😊

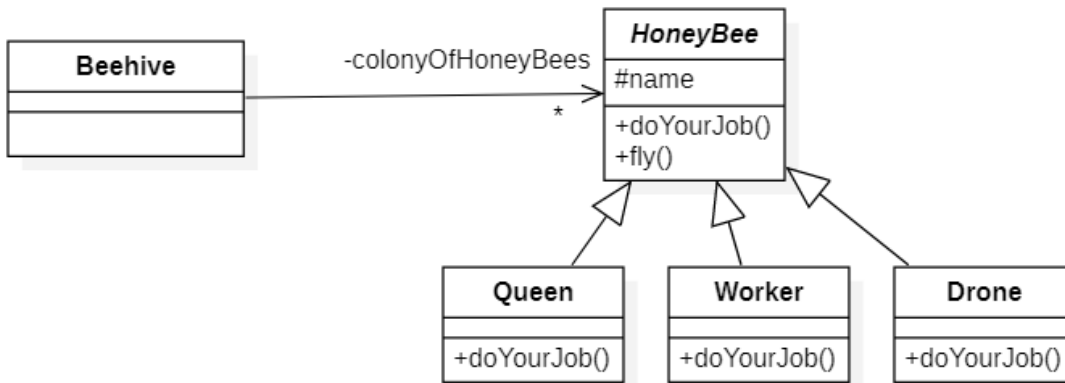
Bien sûr cette ruche devra abriter une colonie d'abeilles qu'il souhaite pouvoir faire voler et faire travailler.

A noter que :

- **ruche** se traduit par le terme anglais **beehive**
- **colonie** se traduit par le terme anglais **colony**
- **hausse à miel** se traduit par le terme anglais **honey super**

Après cette rapide analyse, vous lancez dans la phase de conception. Sachant que vous disposez déjà de la hiérarchie autour de la classe abstraite **HoneyBee**, deux conceptions émergent dans votre tête :

❑ Conception n°1 :

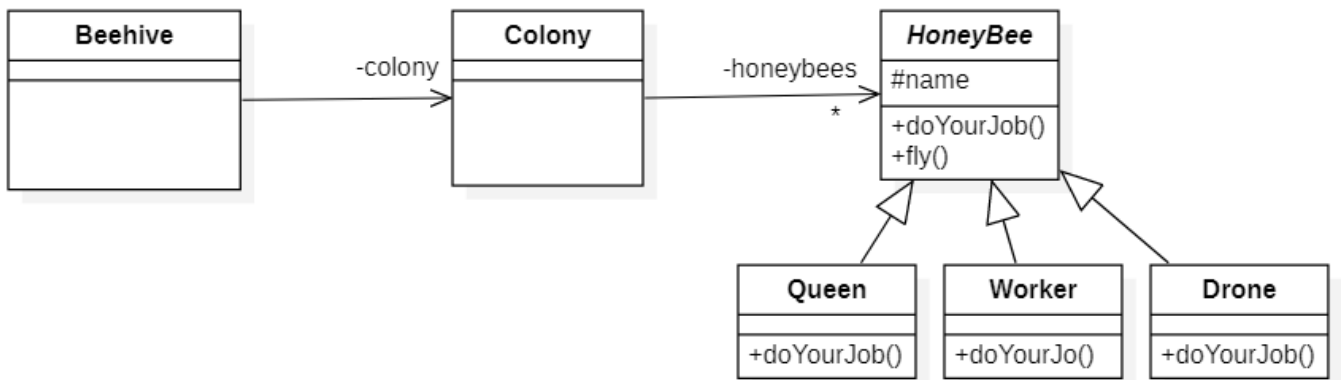


qui donnerait en terme d'implémentation :

```

public class Beehive {
    private List<HoneyBee> colonyOfHoneyBees;
    //...
}
  
```

❑ Conception n°2 :



qui donnerait en terme d'implémentation :

```

public class Beehive {
    private Colony colony;
    //...
}

public class Colony {
    private List<HoneyBee> colonyOfHoneyBees;
    //...
}
  
```

1. Que pensez-vous des conceptions n°1 et n°2 ?
Quelle conception auriez-vous choisie et pourquoi ?
2. **Complétez la conception choisie en reportant les besoins de l'apiculteur, énoncé dans l'analyse, au « bon » endroit** c-a-d dans le « bon » composant : celui qui a bien la responsabilité de gérer tel **attribut** ou d'implémenter tel comportement (**méthode**).



Vous pouvez maintenant passer en **mode connecté** 😊

From HoneyBee to Beehives (Facultatif)

Reprenez le projet **honeybee** et créez une nouvelle classe **BeehivesMain** dans votre package **application** avec une méthode **main** qui sera le point d'entrée du projet pour votre apiculteur ...

A la fin de chaque question, appelez votre enseignant de TP pour qu'il « joue » avec votre code et jette un petit coup d'œil sur sa qualité 😊

1. Un menu pour agrandir la colonie

Pour commencer, l'apiculteur souhaite pouvoir jouer **avec une seule ruche**.

Au démarrage de votre programme, le nom de la ruche et le nombre de hausses de la ruche devront être demandés à l'apiculteur pour pouvoir créer la ruche : la création de la ruche se faisant bien sûr avec une colonie.

La première fonctionnalité que vous développez pour votre apiculteur est un menu qui doit lui permettre d'ajouter une abeille dans sa colonie

La colonie de la ruche <... nom choisi au démarrage...>
contient actuellement 0 abeille(s)

- > 0 reine(s)
- > 0 ouvrière(s)
- > 0 faux-bourdon(s)

-
1. Ajouter une reine
 2. Ajouter une ouvrière
 3. Ajouter un faux-bourdon
 4. Quitter

Dès qu'un ajout est effectué, le menu doit se réafficher. Par exemple, si vous venez d'ajouter une reine, le menu devra maintenant indiquer :

La colonie de la ruche <... nom choisi au démarrage...>
Contient actuellement 1 abeille(s)

- > 1 reine(s)
- > 0 ouvrière(s)
- > 0 faux-bourdon(s)

-
1. Ajouter une reine
 2. Ajouter une ouvrière
 3. Ajouter un faux-bourdon
 4. Quitter

L'apiculteur vous laisse libre cours à votre créativité pour la présentation des étapes devant demander tous les renseignements nécessaires à l'ajout d'une nouvelle abeille dans la colonie 😊

Remarques : ☐ Dans notre monde idéal, on ne fait qu'ajouter des abeilles, on en supprime jamais !
☐ Vous codez en anglais, mais les messages pour votre voisin doivent être en français !
☐ Vous veillerez à la qualité de code : nommage explicite, DRY (Don't Repeat Yourself, décomposer/extraire en sous-méthodes, passer régulièrement le code sous sonarLint,

2. Bug Fix !!!



Fier de votre premier menu, vous demandez à votre apiculteur préféré de venir le tester...

Horreur !!! A la première utilisation : il a trouvé un bug dans votre application !

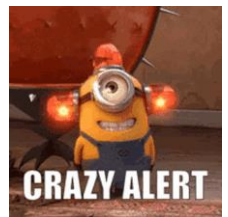
Une colonie d'abeilles ne peut avoir qu'une seule reine et lui, avec votre menu, peut en ajouter plusieurs sans problème.

Vous le rassurez et modifiez votre programme pour corriger ce bug (on parle alors de **fixation de bug** ou **bug fix** !)




3. Un autre petit détail à *fixer* ...

- ☐ L'apiculteur en utilisant votre application a également **remarqué « un autre petit détail à fixer »** en jouant avec votre code.

Il ne souhaite pas donner un **nom** aux abeilles, mais plutôt indiquer une **espèce** d'abeilles. Dans le code **metier** renommer rapidement grâce à l'IDE (menu **Refactor** → **Rename**) N'oubliez pas de remplacer également le mot nom par espèce dans les messages que vous affichez à l'utilisateur !



- ☐ En outre, il vous précise qu'il ne souhaite mettre dans sa ruche que les 3 espèces suivantes qui représente les abeilles d'élevage les plus fréquentes :

		
L'abeille noire (<i>Apis Mellifera Mellifera</i>)	L'abeille italienne (<i>Apis Mellifera Ligustica</i>)	L'abeille caucasienne (<i>Apis mellifera caucasica</i>)

Extraits de <https://www.apiculture.net/blog/les-differents-typesraces-dabeilles-n9>

En fonction de ces derniers détails donnés par l'apiculteur :

- ⇒ Comment envisagez-vous de modifier le typage de l'attribut **espece** ?
Faites apparaître cette modification sur votre diagramme de classe du début d'énoncé.
- ⇒ Implémentez cette modification dans le code **metier**.
Bien sûr, vous ne demanderez pas à l'apiculteur de taper une espèce, vous le ferez choisir une espèce parmi les espèces que vous proposez.

4. Ajout d'une nouvelle option dans votre menu



Vous demandez à votre apiculteur préféré de revenir tester votre menu et là il a une autre petite requête à vous soumettre 😊

Il trouve que saisir les abeilles une à une n'est pas très *user-friendly*.

Il souhaiterait que vous lui proposer de nouvelles options dans votre menu qui lui permettrait plutôt que d'ajouter une seule abeille, d'ajouter d'un seul coup un groupe d'abeilles (après tout 1 abeille n'est-ce pas un groupe composée d'une seule abeille 😊)...
N'oubliez pas qu'il n'y a pas de groupe « reine », on ne peut ajouter qu'une seule reine à la fois !!!

☐ D'un commun accord, vous décidez que le menu doit être modifié de la manière suivante :

La colonie de la ruche <... **nom choisi au démarrage**...>
contient actuellement 0 abeille(s)

- > 0 reine
- > 0 ouvrière(s)
- > 0 faux-bourdon(s)

-
1. Ajouter une reine
 2. Ajouter un groupe d'ouvrières
 3. Ajouter un groupe de faux-bourdons
 4. Quitter

L'apiculteur vous laisse libre cours à votre créativité pour la présentation des étapes devant demander tous les renseignements nécessaire à l'ajout d'une nouvelle abeille dans la colonie 😊

☐ Au fait, à quoi cela sert-il de laisser l'option 1 : Ajouter une reine si une reine a déjà été ajoutée. Ainsi si une reine est présente dans le colonie, il sera de bon ton de rendre cette option pour l'apiculteur :

La colonie de la ruche <... **nom choisi au démarrage**...>
contient actuellement 0 abeille(s)

- > 1 reine
- > 0 ouvrière(s)
- > 0 faux-bourdon(s)

-
2. Ajouter un groupe d'ouvrières
 3. Ajouter un groupe de faux-bourdons
 4. Quitter

Bon cela n'est pas très joli de commencer par l'option 2... Une petite astuce, consiste à renommer les options 😊

Donc maintenant, à la **création de la ruche**, votre menu doit ressembler à :

```
-----  
La colonie de la ruche <... nom choisi au démarrage...>  
contient actuellement 0 abeille(s)  
-> 0 reine  
-> 0 ouvrière(s)  
-> 0 faux-bourdon(s)  
-----
```

1. Ajouter un groupe d'ouvrières
2. Ajouter un groupe de faux-bourdons
3. Ajouter une reine
0. Quitter

Tant que la ruche ne possède pas de reine, l'option **ajouter une reine** doit être disponible.
Dès que la ruche possède une reine, l'option ne doit plus être accessible et le menu devient alors :

```
-----  
La colonie de la ruche <... nom choisi au démarrage...>  
contient actuellement 0 abeille(s)  
-> 0 reine  
-> 0 ouvrière(s)  
-> 0 faux-bourdon(s)  
-----
```

1. Ajouter un groupe d'ouvrières
2. Ajouter un groupe de faux-bourdons
0. Quitter

☐ Optionnel : Il est même possible d'aller même plus loin et de forcer l'utilisateur à entrer une reine dès le départ car que se passe-t-il si une ruche n'a pas de reine ?

5. Menu pour suivre l'évolution de sa ruche

L'apiculteur souhaite maintenant pour voir interagir avec sa ruche.

Vous décidez d'un commun accord que le menu qui doit s'afficher dorénavant est le suivant :

```
-----  
Bienvenue dans la ruche <... nom choisi au démarrage...>  
Actuellement, cette ruche possède <nb choisi> hausses  
et la colonie contient 0 abeille(s)  
-----
```

1. Ajouter une hausse
2. Agrandir la colonie
3. Détailler la colonie
0. Quitter

⇒ Lorsque, vous appellerez l'option **2.Agrandir la colonie**, vous utiliserez le comportement que vous avez implémenté précédent. Vous changerez toutefois l'intitulé de l'option 0 du menu précédent qui ne sera plus 0. Quitter, mais plutôt : **0.Retourner au menu précédent**

Veillez bien à la qualité de code : nommage explicité, DRY (Don't Repeat Yourself, décomposer/extraire en sous-méthodes, passer régulièrement le code sous sonarLint,)

Pour que votre code soit facilement lisible, vous devez le décomposer : pensez au menu **Refactor - > Extract Method** qui vous sera sûrement d'une grande aide !!!
(pas de méthode de 50 lignes !!!)

⇒ Lorsque, vous choisirez l'option **3.Détailler la colonie**, l'apiculteur souhaiterait visualiser un menu du genre

```
-----  
Bienvenue dans la ruche <... nom choisi au démarrage...>  
Actuellement, la colonie de cette ruche  
contient actuellement 29 abeille(s)  
-----
```

- > 1 reine (abeille noire)
- > 25 ouvrière(s)
 - > 20 abeille(s) noire(s)
 - > 10 abeille(s) italienne(s)
- > 3 faux-bourdon(s)
 - > 3 abeille(s) noire(s)

6. Faire « vivre la ruche »

L'apiculteur souhaite ensuite pouvoir faire « vivre sa ruche » c.-à-d. pouvoir choisir de faire voler ou de faire travailler sa colonie d'abeilles.

D'un commun accord, vous proposez le menu suivant :

Bienvenue dans la ruche <... nom choisi au démarrage...>
Actuellement, cette ruche possède <nb choisi> hausses
et la colonie contient 29 abeille(s)

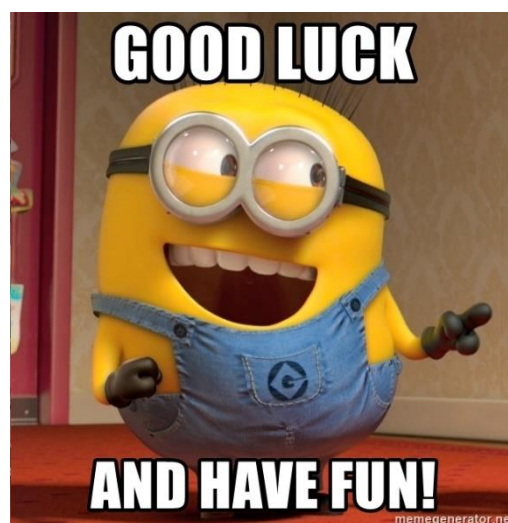
1. Ajouter une hausse
2. Agrandir la colonie
3. Détailler la colonie
4. Faire voler la colonie
5. Faire travailler la colonie
0. Quitter

Remarque : Les options suivantes n'apparaîtront uniquement si une abeille est présente dans la colonie :
Faire voler la colonie
Faire travailler la colonie

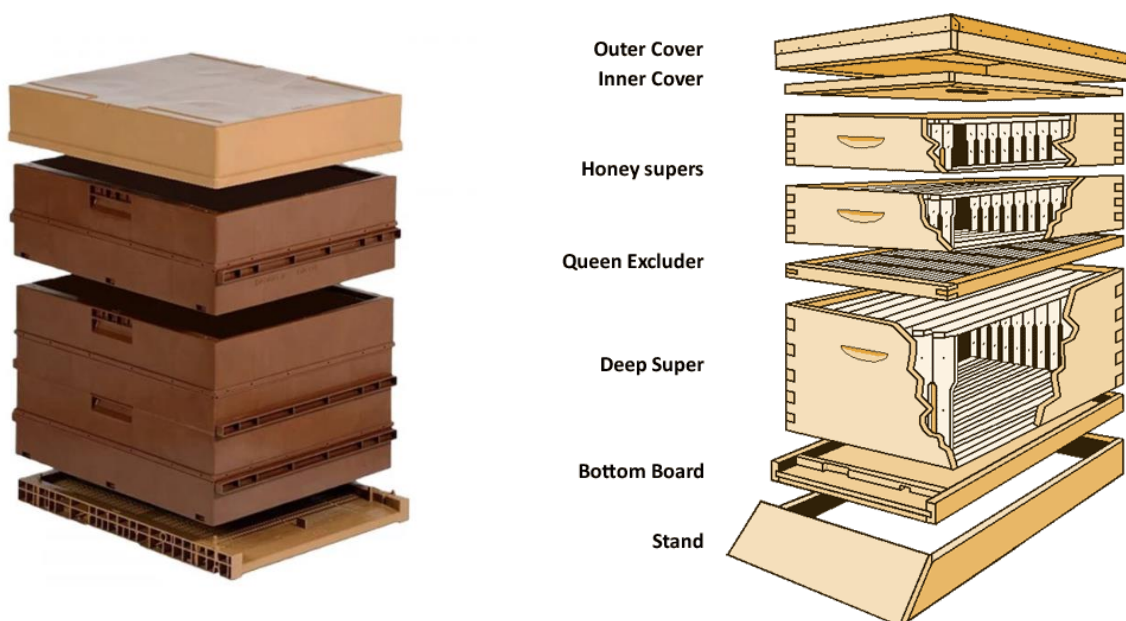
7. Passage à l'échelle : faire « vivre » autant de ruches que l'on souhaite !

L'apiculteur est ravi de votre programme. Cela fait maintenant une semaine qu'il fait vivre sa seule et unique ruche et il souhaiterait passer à la vitesse supérieure. En réalité, il souhaiterait disposer d'au moins 3 ruches. Il vous demande donc de lui faire évoluer le programme pour qu'il puisse maintenant jouer avec autant de ruches qu'il le souhaite !!!!

Et vous pouvez continuer à imaginer d'autres options pour rendre votre programme encore plus interactif : par exemple supprimer des abeilles de la colonie ou supprimer des hausses ou toute autre fonctionnalité créative que vous pouvez imaginer pour rendre votre programme plus sympathique ... 😊



Annexe : Zoom sur les éléments de la ruche de notre apiculteur



Une **ruche complète économique** et durable construite en France par le célèbre fabricant Nicotplast en **plastique alimentaire recyclable** conforme aux normes européennes (sans bisphénol A ni phtalates) !

La **ruche Nicot** est une ruche Dadant légère et facilement transportable qui est particulièrement adaptée à la **transhumance**.

Cet habitat offre aux abeilles un nid confortable et à l'apiculteur les compartiments nécessaires à la production de miel : corps, hausse, fond ventilation totale et couvre-cadre transparent.



Documents extraits de :

<https://www.apiculture.net/ruches-corps-et-hausse-nicot/5896-ruche-nicot-dadant-10-cadres-sans-cadres.html>

<https://backyardbeekeeping101.com/when-to-add-honey-supers/>