TP 3 : modélisation, premières classes

Exercice 1 (Échauffement)**.** – Consulter la documentation de la fonction `Math.random`. Pour cela, faites une recherche sur le web, avec les mots clefs `java API Math.random`. Remontez sur la page principale de la documentation de l'API de java. Gardez la toujours sous le coude.
- Téléchargez l'archive indiquée sur la page web du cours dans le répertoire ~/CCI–LO/TP3/.
- Lancer Eclipse, choisissez comme espace de travail ~/CCI–LO, et créez un nouveau projet TP3 en configurant ~/CCI–LO/TP3 comme répertoire source. Voir par exemple le document sur Eclipse sur la page web de cours pour de l'aide.
- Parcourez les exemples du cours et essayez ceux dont vous ne devinez pas immédiatement le comportement.

Exercice 2 (Auto-quizz)**.**
Qu'est-ce qu'un(e)
- attribut ?
- attribut statique (appelé aussi : attribut de classe) ?
- méthode ?
- méthode statique (appelée aussi : méthode de classe) ?

Exercice 3 (Une première classe : modèle de Lotka-Volterra)**.**
On souhaite faire une simulation numérique pour un système écologique simple composé de deux populations de proies (X) et de prédateurs (Y) de tailles respectives $x(t)$ et $y(t)$. À chaque pas de temps, les populations évoluent selon les règles suivantes :
- Le nombre de naissances dans la population X est proportionnel à $x(t)$
- Le nombre de décès dans la population X est proportionnel au produit $x(t)y(t)$
- Le nombre de décès dans la population Y est proportionnel à $y(t)$.
- Le nombre de naissances dans la population Y est proportionnel au produit $x(t)y(t)$.

On note α, β, γ, δ les paramètres donnant les coefficients de proportionnalité.

(1) Quelle est la signification des règles de ce modèle ?

(2) Exprimer, en fonction de $x(t)$ et $y(t)$ et des paramètres la valeur de $x(t+1)$ et de $y(t+1)$.

(3) Implanter une classe `LotkaVolterraModele` en complétant le squelette fourni. Une instance de cette classe représentera l'état d'un modèle à un moment donné. Elle aura donc 7 attributs : t, x, y, les constantes. À chaque appel de la méthode évolue, t sera incrémenté de 1 et x et y seront mis à jour.

(4) Lancer l'application `ModeleLotkaVolterraGUI` pour visualiser le résultat de la simulation numérique. Sont affichées :
- En vert : la courbe de l'évolution de la population $x(t)$
- En rouge : la courbe de l'évolution de la population $y(t)$

Échelle horizontale : t entre 0 et 200. Échelle verticale : population entre 0 et 1000 ;

Les glissières permettent de contrôler respectivement les valeurs initiales $x(0)$ et $y(0)$, ainsi que les valeurs des quatre paramètres $\alpha, \beta, \gamma, \delta$.

(5) Expérimentez avec ces paramètres pour voir quels sont les scénarios possibles pour l'évolution de la population.

(6) Pour en savoir plus : `http://fr.wikipedia.org/wiki/%C3%89quations_de_Lotka-Volterra`

Exercice 4 (Un premier animal)**.**
L'objectif est d'implanter un petit animal virtuel type Tamagotchi. Un tamagotchi a deux opération : `se_lave` et `attend`.

L'état d'un tel animal est modélisé par un attribut de type `int` mesurant sa propreté. S'il est trop sale, c'est-à-dire si l'attribut tombe à zéro, l'animal meurt. S'il se lave trop aussi.

(1) Fixer les paramètres : combien de points de propreté un tamagotchi a-t'il au départ ; combien il en gagne et perd lorsqu'il se lave ou attend, etc.

(2) Écrire quelques scénarios de tests qui illustrent le comportement d'un tamagotchi ; du genre :

```
Tamagotchi t = new Tamagotchi();
t.toString();                    -> "Un tamagotchi"
t.se_lave();
t.toString();                    -> "Un tamagotchi propre"
```

(3) En vous inspirant de la classe `Souris`, implanter une classe `Tamagotchi`. Chaque paramètre sera défini dans une constante en début de classe du genre :

```
    static final PROPRETE_INITIALE = 5;
```

(4) Vérifier que votre classe implante correctement tous vos scénarios de tests.