

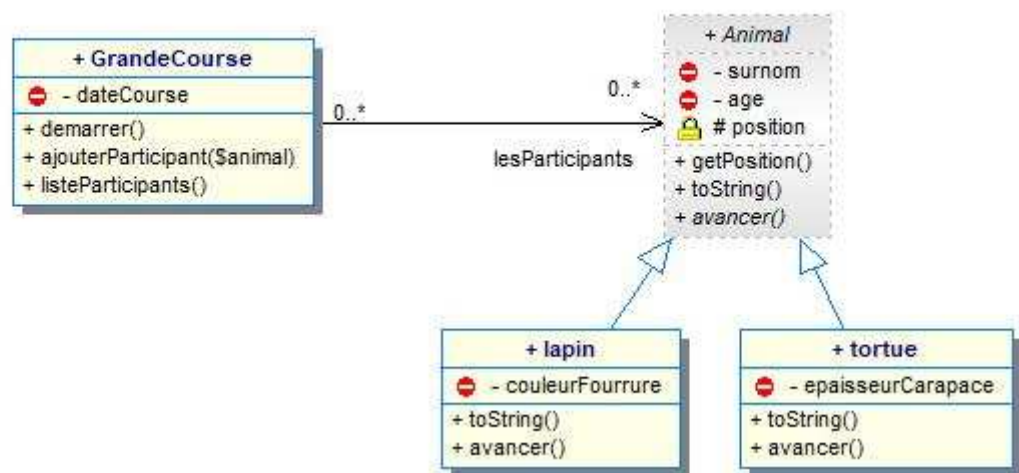
# TP3 Mise en Application des concepts :

## Héritage, redéfinition, abstraction, polymorphisme et statique

### Thème : La grande course

Pour sa nouvelle édition, notre grande course de lapins a décidé de mettre un peu de piquant dans la compétition en invitant des tortues et la course devient, de fait, la grande course des animaux.

Dans notre nouveau modèle, les lapins possèdent les caractéristiques suivantes : surnom, age, couleur de la fourrure et position (un nombre entier qui mesure la distance depuis la ligne de départ). Les tortues possèdent quant à elle les caractéristiques suivantes : surnom, age, épaisseur de la carapace et position. Les deux animaux possèdent les méthodes Avancer et ToString (représenter l'animal sous forme de chaîne de caractères). On constate immédiatement qu'une tortue et un lapin sont des sortes d'animaux qui possèdent des éléments en commun, nous allons donc créer une classe Animal qui regroupe tout ce qui est commun. Nous pourrons alors écrire deux classes : Lapin et Tortue qui héritent de Animal. Chacune de ces classes ajoutera les caractéristiques propres à l'animal et redéfinira les méthodes de manières adéquates.



classes fournies : toutes les classes sont données, à vous de les compléter !

## PARTIE 1 : héritage et redéfinition et abstraction

1. Écrivez la classe `Animal` contenant tous les éléments commun aux lapins et aux tortues. Écrivez un constructeur paramétrique permettant de facilement initialiser les propriétés. La méthode `string ToString()` retourne une chaîne de caractère contenant les informations sur l'animal : « surnom, age, position ». La méthode `void Avancer()` est connue des lapins et des tortues mais l'action à réaliser n'est pas déterminée pour la classe `Animal` que peut-on observer sur cette méthode dans la classe `Animal` ? Quelle conséquence sur le classe ?
2. Écrivez les classes `Lapin` et `Tortue` qui héritent de `Animal`. Ajoutez les champs nécessaires et définissez des constructeurs paramétriques pour les initialiser.
3. Redéfinissez la méthode `ToString` dans les classes `Lapin` et `Tortue` afin d'ajouter les informations spécifiques à la description de l'animal en question. Par exemple, l'appel de `ToString` sur un objet de type `Tortue` renvoie une chaîne de la forme « surnom, age, position, épaisseur de la carapace ».
4. Redéfinissez (obligatoire ici car abstraite dans la classe mère) la méthode `Avancer` dans les classes `Lapin` et `Tortue`.

Les lapins, ***consultant leur portable à chaque notification perdent du temps et sont déconcentrés***, la méthode `Avancer` fait avancer le lapin d'une distance aléatoire entre 0 et 10.

La tortue, ***concentrée et ne se laissant pas distraire par des évènements extérieurs***, avance toujours de 6 lorsque l'on appelle la méthode `Avancer`.

La méthode d'instance `Random_int(int min, int max) : int` de la classe `Random_int` génère des nombres entiers pseudo-aléatoire cryptographiquement sécurisé..

5. Dans une page Web nommée `Index.php` :  
Créer un animal générique nommé "monAnimal", de 2 ans. Est-ce possible ? Quel message vous indique-t-il ?  
Créer 3 lapins et deux tortues et les ranger dans un tableau d'animaux. Faites avancer (*utilisation de la redéfinition de la méthode `avancer()`*) chacun des animaux **100 fois (pour les tests et les captures d'écran, j'ai mis que 5 fois)**, visualiser l'avancement de chaque animal (*utilisation de la redéfinition de la méthode `__toString()`*)!  
Déterminez quel animal a remporté la course et afficher ses caractéristiques.

Exemple d'exécution sur 5 fois :

\*\*\*\*\* PARTIE 1 du TP : : héritage et redéfinition et abstraction\*\*\*\*\*

participant n°1

jeannot - 1 ans à la position 4 couleur blanc

jeannot - 1 ans à la position 4 couleur blanc

jeannot - 1 ans à la position 12 couleur blanc

jeannot - 1 ans à la position 18 couleur blanc

jeannot - 1 ans à la position 21 couleur blanc

participant n°2

lapinette - 2 ans à la position 10 couleur grisperle

lapinette - 2 ans à la position 10 couleur grisperle

lapinette - 2 ans à la position 13 couleur grisperle

lapinette - 2 ans à la position 16 couleur grisperle

lapinette - 2 ans à la position 26 couleur grisperle

participant n°3

robin - 24 ans à la position 3 couleur fauve

robin - 24 ans à la position 4 couleur fauve

robin - 24 ans à la position 9 couleur fauve

robin - 24 ans à la position 17 couleur fauve

robin - 24 ans à la position 17 couleur fauve

participant n°4

lenteure - 38 ans à la position 6 carapace 3

lenteure - 38 ans à la position 12 carapace 3

lenteure - 38 ans à la position 18 carapace 3

lenteure - 38 ans à la position 24 carapace 3

lenteure - 38 ans à la position 30 carapace 3

le GAGNANT de la COURSE EST...lenteure - 38 ans à la position 30 carapace 3

## **PARTIE 2 : collection et polymorphisme**

6. Dans la classe GrandeCourse, créer son constructeur en précisant la date de la course
7. Créer la méthode ajouterParticipant() qui permet d'ajouter un animal à la course.
8. Créer la méthode listeParticipants qui retourne une chaine contenant la liste des animaux participants mais en en précisant si c'est un lapin ou une tortue (principe du polymorphisme : à tester avec instance of)

\*\*\*\*\* PARTIE 2 du TP : : collection et polymorphisme \*\*\*\*\*

liste des participants à la Grande Course

LAPIN n°1

jeannot - 1 ans à la position 21 couleur blanc

LAPIN n°2

lapinette - 2 ans à la position 26 couleur grisperle

LAPIN n°3

robin - 24 ans à la position 17 couleur fauve

TORTUE n°4

lenteure - 38 ans à la position 30 carapace 3

9. Créer la méthode démarrer() qui fait avancer chaque participant (utilisation de la redéfinition de la méthode avancer())
10. Où faut-il créer la méthode Terminer() pour déterminer qui est le vainqueur de cette Grande Course ! Cette méthode retournera le participant vainqueur de cette course !
11. Compléter (partie 2) le fichier index.php pour prendre en compte la classe GrandeCourse (instancier cette classe, y ajouter les 4 participants (déjà instanciés en partie 1), afficher tous les participants et démarrer la course et la terminer pour afficher le vainqueur)

#### DEMARRAGE DE LA COURSE...

participant n°1

jeannot - 1 ans à la position 30 couleur blanc

jeannot - 1 ans à la position 32 couleur blanc

jeannot - 1 ans à la position 37 couleur blanc

jeannot - 1 ans à la position 38 couleur blanc

jeannot - 1 ans à la position 41 couleur blanc

participant n°2

lapinette - 2 ans à la position 28 couleur grisperle

lapinette - 2 ans à la position 38 couleur grisperle

lapinette - 2 ans à la position 48 couleur grisperle

lapinette - 2 ans à la position 49 couleur grisperle

lapinette - 2 ans à la position 50 couleur grisperle

participant n°3

robin - 24 ans à la position 20 couleur fauve

robin - 24 ans à la position 26 couleur fauve

robin - 24 ans à la position 33 couleur fauve

robin - 24 ans à la position 42 couleur fauve

robin - 24 ans à la position 48 couleur fauve

participant n°4

lenteure - 38 ans à la position 36 carapace 3

lenteure - 38 ans à la position 42 carapace 3

lenteure - 38 ans à la position 48 carapace 3

lenteure - 38 ans à la position 54 carapace 3

lenteure - 38 ans à la position 60 carapace 3

le GAGNANT de la COURSE 1 EST...lenteure - 38 ans à la position 60 carapace 3

**PARTIE 3 : notion STATIQUE et Nouvelle Course avec les mêmes participants (remise à zéro de leur position)**

12. Que faut-il faire pour que les participants puisse faire recommencer la course avec la remise à zéro de la position des participants Où faut-il le faire ?
13. Faites les modifications nécessaires dans les classes et dans la page Index.php
14. Faites 3 courses avec les mêmes participants (avec une variable statique déterminer le numéro de la course) et afficher le GRAND VAINQUEUR !

liste des VAINQUEURS AUX 3 Courses

TORTUE n°1

lenteure - 38 ans à la position 600 carapace 3

TORTUE n°2

lenteure - 38 ans à la position 600 carapace 3

TORTUE n°3

lenteure - 38 ans à la position 600 carapace 3

le GRAND GAGNANT des 3 courses est lenteure - 38 ans à la position 600 carapace 3

**LA MORALE de cette histoire :**

Mieux vaut être plus lent mais concentré !