Prétest 1 OCI : Les bases de la programmation orientée objet

Problème 1

La figure ?? donne le diagramme de classe des acteurs du scénario leaves-and-wombats. On donne également les en-têtes des classes figurant dans ce diagramme à l'exception de la classe `Actor`.

```
public class Leaf extends Actor
public class Wombat extends Actor
```

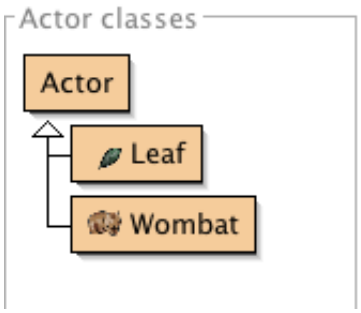

FIGURE 1 – Le diagramme de classe du scénario leaves-and-wombats

Dessiner ci-dessous le diagramme de classe des acteurs du scénario sports à partir des en-têtes de classes donnés ci-dessous :

```
public class Football extends TeamGame
public class TeamGame extends Game
public class IndividualGame extends Game
public class BeachFootball extends Football
public class Bike extends IndividualGame
```

Problème 2

Écrire les *en-têtes* des méthodes dont les caractéristiques sont données ci-dessous.

a) La méthode `produit` prend 2 paramètres de type entier et renvoie une valeur entière.

b) La méthode `cube` prend 1 paramètre de type entier et renvoie une valeur entière.

c) La méthode `isHappy` ne prend aucun paramètre et renvoie une valeur booléenne.

d) La méthode `displayIntegerValue` prend 1 paramètre de type nombre entier et ne renvoie aucune valeur.

Problème 3

a) La variable `gunReloadTime` de la classe `Rocket` fixe la cadence de tir d'une fusée. Écrire une instruction qui donne la valeur 6 à cette variable.

b) On dispose de la méthode de signature `void total(int x, int y)`.
Écrire un appel de cette méthode.

c) Considérons l'instruction `boolean b = isOdd(5);`.
Écrire la signature de la méthode `isOdd`.

d) Que renvoie la méthode `displayPoints` dont la signature est donnée ci-dessous ?
`void displayPoints()`.

Problème 4

Dans la méthode `act` de la classe `Wombat` du scénario leaves-and-wombats, on utilise les méthodes `foundLeaf`, `eatLeaf` et `setDirection` (cf. résumé des méthodes de la classe `Wombat` en page **??**) .
Le compilateur Java indique une (des) faute(s) dans le code source donné ci-dessous. Ecrire à nouveau cette méthode `act` en éliminant les fautes de syntaxe.

```
import greenfoot.*;

public class Wombat extends Actor
{
   public void act()
   {
     if (foundLeaf)
      {
        eatLeaf()
        setDirection();
      }
    }
}
```

Méthode `act` avec une syntaxe correcte :

Problème 5

Les réponses aux questions de cet exercice sont à rédiger sur la page suivante. On donne ci-dessous les méthodes de la classe `Wombat` du scénario leaves-and-wombats.

Constructor Summary

`Wombat()`

Method Summary

void	**act**() Do whatever the wombat likes to to just now.
boolean	**canMove**() Test if we can move forward.
void	**eatLeaf**() Eat a leaf (if there is one in our cell).
boolean	**foundLeaf**() Check whether there is a leaf in the same cell as we are.
int	**getLeavesEaten**() Tell how many leaves we have eaten.
void	**move**() Move one step forward.
void	**setDirection**(int direction) Set the direction we're facing.
void	**turnLeft**() Turn left by 90 degrees.

Methods inherited from class greenfoot.Actor

addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

a) Que renvoie la méthode `getLeavesEaten` ? Donner son type.

Pour **chaque question ci-dessous**, écrire la méthode `act` de de la classe `Wombat` qui hérite de la classe `Actor`, en tenant compte des conditions :

b) Si le wombat peut avancer, il avance d'un pas et ensuite il tourne à gauche.

c) Si le wombat trouve une feuille, il la mange. Ensuite, et seulement dans le cas où il trouve une feuille, si c'est possible, il avance d'un pas et si c'est encore possible, il avance d'un deuxième pas.

d) Si le wombat peut avancer, il avance d'un pas et ensuite il tourne **à droite** de 90 degrés.

e) Si le wombat ne trouve pas une feuille, il prend la direction *est (droite)* ; s'il en trouve une, il prend la direction *sud (bas)*.

Rappel : les quatre directions possibles sont codées avec les entiers 0 ,1, 2 et 3 qui correspondent respectivement aux directions est, sud, ouest, nord