

TP, deuxième série d'exercices

Benoît Valiron
benoit.valiron@lipn.univ-paris13.fr

22 novembre 2010

Introduction

Encore une fois, vous êtes sensés me rendre ce que vous avez réalisé.

Exercice 1.

Dans un dossier `Exo1`, reprenez la classe `Point`.

1. Ajoutez les méthodes `public double getX()`, `public double getY()` à la classe `Point`.
2. Reproduisez les deux sous-classes `Point3D` et `PointCouleur` du cours.
3. Modifiez la méthode `toString` de chacune d'entre elle pour prendre en compte les informations supplémentaires qu'elles contiennent.
4. Dans une classe `TestPoint`, placez une méthode `main` :

```
public class TestPoint {  
    public static void main(String[] args){  
        // Placez votre code ici  
    }  
}
```

et définissez deux variables de type `Point`. Dans la première, placez un objet de type `Point3D` et dans la deuxième un `PointCouleur`. Affichez les deux points à l'écran à l'aide de `System.out.println`.

5. Créez deux interfaces :
 - `InterfacePoint` qui contient les méthodes `public double getX()`, `public double getY()` et `public void translate(double vx, double vy)`.
 - `InterfaceCouleur` qui contient une méthode `public void changeCouleur(String c)`.
6. Modifiez les classes des questions 1 et 2 pour implémenter les interfaces créés : Quelles classes implémentent ces interfaces ?
7. Changez la portée des champs, constructeurs et méthodes pour être le plus restrictif possible (mais pour que ça compile).

Exercice 2.

Dans un répertoire `Exo2`, copiez la classe `TamponEntiers`, et changez les portées : les champs `index` et `tableau` ont-ils besoin d'être accessibles ?

1. Créez une interface `GenFonction` avec une méthode `int fun(int x)`.
2. Créez deux classes `Double` et `Flip` qui implémentent `GenFonction`. Elle n'ont pas de champs. La première classe possède une fonction qui envoie `x` sur `2*x` et la deuxième rends 1 si `x==0` et 0 sinon.
3. Créez une classe `OutilsFonction` avec
 - un champ `GenFonction f`,
 - un constructeur qui initialise le champ,
 - une méthode `TamponEntiers iteration(int n, int x)` qui calcule `x`, `f.fun(x)`, `f.fun(f.fun(x))`, ..., `n` fois, et range la séquence dans un objet de type `TamponEntiers`, puis le retourne.
4. Créez une classe `Test` avec une méthode `main`. Instanciez un objet `o` de type `GenFonction`. Placez un objet de type `Double` dans `o`. Calculez les itérés correspondant à `o.fun` avec la méthode `iteration`. Choisissez 15 itérations, et commencez avec `x = 2`. Imprimez les résultats.
5. Comment feriez vous pour avoir les itérés de la méthode `fun` de la classe `Flip` ?