Architecture Logicielle – TP 04 2022–2023

Le patron *Composite*

Exercice 1 – *Mise en place d'un composite.*

Pour cette partie, nous repartirons du projet du TP N°3 (*Les patrons de construction – partie 2*).

On souhaite ajouter le concept de *sous-image* (*subpicture*) à la liste de nos composants graphiques. Une sous-image correspond à l'opération « grouper » dans un logiciel de dessin vectoriel et permet donc d'assembler un certain nombre de formes déjà existantes et de les traiter comme un seul objet (une seule forme). Des formes pourront être ajoutées dynamiquement à la sous-image. Les opérations communes à toutes les formes devront s'appliquer également aux sous-images. Ici, nous avons l'opération de dessin *draw()*, que nous avons gardée... pour l'instant. Pour rendre les choses plus intéressantes, on propose d'ajouter à tous les types de formes (donc dans l'interface `Drawable` [1]) l'opération `move(int dx, int dy)` qui translate selon le vecteur (dx, dy) la forme sur laquelle elle est appliquée.

Dans un premier temps, pour simplifier l'implémentation des opérations, on utilisera une approche récursive où les composites délégueront à leurs composants les traitements à réaliser.

Question 1.1 : Rappelez le diagramme UML de principe du patron sans regarder dans le cours (ni sur Internet ;-)).

Question 1.2 : Donnez le diagramme UML de votre solution. Identifiez tous les acteurs du patron en annotant [2] vos entités et rédigez complètement le contenu de vos interfaces.

Question 1.3 : Rédigez précisément le contenu de l'interface commune aux feuilles et aux nœuds.

Question 1.4 : Rédigez précisément le contenu de l'interface publique implicite (ou pas ;-)) de votre composite.

Question 1.5 : Codez votre solution.

Question 1.6 : Écrivez une fonction de test `getDemoGroups()` en dupliquant le code de `getDemo()` : vous y créerez une sous-image pour la maison, une seconde sous-image pour le soleil et une troisième sous-image pour le personnage. Testez votre code.

Question 1.7 : Il pleut [3]. Utilisez la puissance du composite pour mettre le personnage à l'abri dans la maison à l'aide d'une unique instruction réalisant un déplacement $(-400, 0)$.

1. Que vous avez probablement nommé `Shape` maintenant suite à votre refactoring
2. Rappel : vos classes ne doivent pas être nommées en fonction des acteurs d'un patron mais d'après leur rôle métier/applicatif. Vous annoterez votre diagramme en écrivant, à coté des classes, le nom de l'acteur correspondant dans le patron.
3. Inutile de dessiner un nuage devant le soleil ... je vous vois venir !