Design Patterns - Laboratoire 1

Laurent Novac, Tibor Udvari, Jacot Niels

L’implémentation du pattern Bridge permet de pouvoir changer dynamiquement les librairies utilisé pour dessiner les primitives cercles et lignes. Le pattern Bridge permet d’avoir un couplage faible entre les formes que l’on veut dessiner et l’implémentation de leur algorithme de dessin (dans notre cas pour les dessins de primitives).   
  
Pour résumer, les méthodes Shape::drawLine() et Shape::drawCircle() appellent les methodes AbstractDrawingStrategy::drawLine() et AbstractDrawingStrategy::drawCircle().   
Mais pour les appeler Shape utilisent son attribut drawingStrategy qui est initialisé dans le constructeur avec soit un V1Drawing ou un V2Drawing qui sont des sous-classes de AbstractDrawingStrategy.   
  
C’est dans ces deux classes qu’on appelle effectivement les méthodes de dessin de GeomGlut.  
Le passage de V1Drawing à V2Drawing peut bien sûr être fait dynamiquement. Pour le démontrer nous instancions soit l’un soit l’autre dans les différentes formes (à la construction).  
  
Le but de tout ça est que les classes Circle, Rectangle, Carree et Triangle puisse appeler leur méthodes drawLine et drawCircle sans se soucier de la librairie utilisé.  
  
En ce qui concerne l’implémentation sans utiliser de pattern, nous avons simplement suivi le canevas du cours.   
  
Pour l’implémentation du Singleton nous avons modifier la classe GeomGlut fourni pour la transformer en Singleton (dans un autre projet). Ainsi il ne sera plus nécessaire d’avoir un attribut externe globale nulle part il suffit d'appeler la méthode statique getInstance() de GeomGlut. Elle s’occupe de vérifier que l’instance est unique et nous donne accès à toutes les méthodes.