

Design Pattern - Labo 1 – 15.03.2015

Auteurs

Horia Mut / Quentin Jeanmonod / Lukas Bitter

But

Appliquer le design pattern Bridge dans une application de dessin.

Description

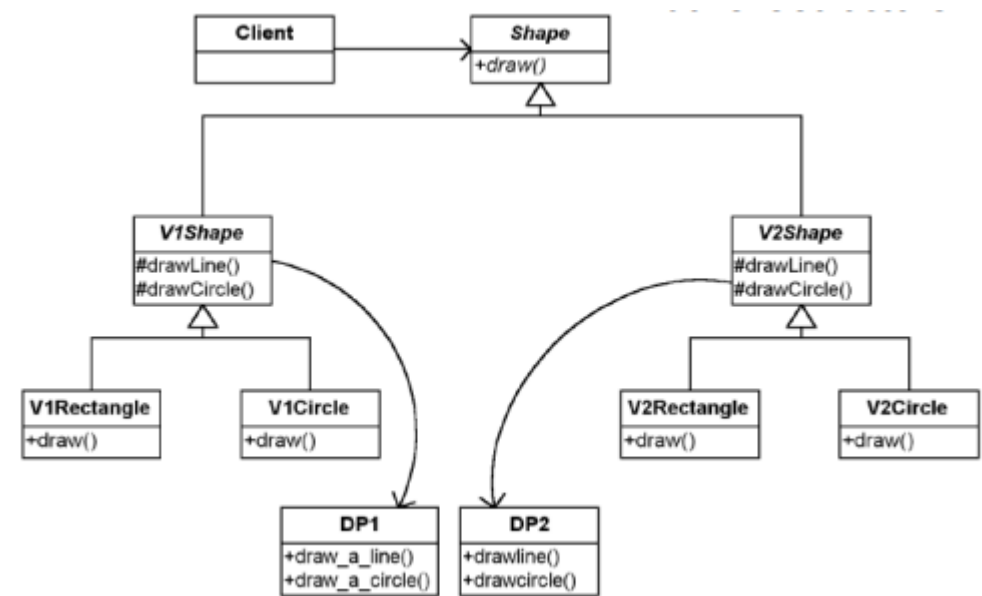
L'application de dessin doit dessiner différentes formes, p. ex rectangles et cercles. Il existe deux ou plus manières différentes de faire ces dessins. Ces manières différentes sont implémentées dans des classes DP1 et DP2, sous forme d'API.

Le client, qui va instancier les formes et appeler leurs fonctions de dessin, sait quelles API seront utilisées pour chaque forme.

V1 sans bridge

Les différentes versions d'API (DP1 et DP2) sont référencées par les formes abstraites V1Shape et V2Shape. Les méthodes des formes abstraites appellent les méthodes des DP1 et DP2.

Ainsi, puisque les formes V1Circle et V1Rectangle dérivent des formes abstraites, elles ont la possibilité d'appeler les méthodes protégées des formes abstraites et ainsi se dessiner.



V1 Avec bridge

Avec le bridge, il n'y a plus besoin de créer des classe de formes ni de formes abstraites pour les différentes versions d'API.

Dans ce modèle, la classe abstraite de formes est composée de références à la classe abstraite de dessin. Comme cette classe de dessin est implémentée par les deux classes des différentes version de dessins et que les classes de formes concrètes implémentent la classe abstraite des formes, les formes concrètes peuvent appeler directement les méthodes des classes des versions, car la

référence dans la classe de forme abstraite a été créée avec la bonne version, passée en paramètre aux formes concrètes lors de la construction.

Cela permet ainsi d'économiser la création de classes de formes concrètes par version. Une seule suffit.

