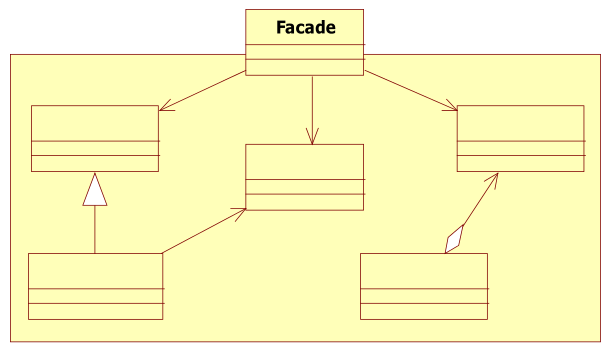
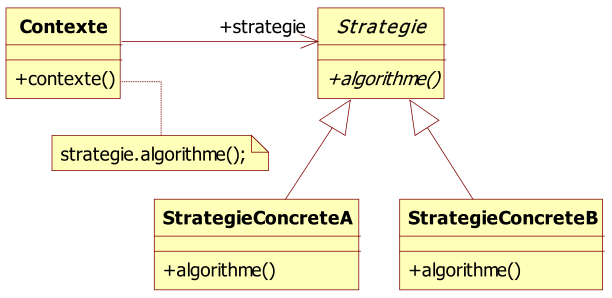
Description de L’architecture :

Patrons de conception utilisés :

Nous avons utilisé une façade (Manager) pour faire le lien entre la vue et le métier sans que la vue ne connaisse le métier, cela est aussi plus pratique car nous donnons le manager à toutes nos fenêtres pour qu’elles puissent accéder aux différentes listes par exemple.



Nous avons utilisé une stratégie au niveau de notre persistance, cela nous permet de pouvoir créer autant de manière différente de persistance que l’on veut. Par exemple nous avons une persistance de base, et une persistance qui utilise des fichiers xml. De plus, si nous créons une nouvelle persistance, nous n’aurons pas à changer les méthodes qui se servent d’une persistance car nous aurons défini une interface, qui pourra donc être n’importe la quelle des persistances.

Dépendances :

Nous avons découpés notre projet en 7 sous-projets :

**-PcComparator.Class :**

Ce projet contient le métier de notre application. Il n’a de référence sur aucun des projets car il n’a pas besoin de les connaitre.

**- PcComparator.Façade :**

Ce projet contient notre Manager. Nous avons choisi de le mettre à part pour bien montrer le rôle d’une façade qui consiste à faire l’intermédiaire entre la vue et le métier sans que l’un ne connaisse l’autre. Il a des références sur : **PcComparator.Class**et **PcComparator.Persistance** car Il a besoin d’une persistance car c’est lui qui instancie la bibliothèque (qui a besoin d’être chargée avec une persistance) et car à l’intérieur de ce projet, on manipule des composants.

**- PcComparator.Persistance :**

Ce projet contient les différents types de persistances. Il a une référence sur **PcComparator.Class** car il y un stub qui instancie une liste de composants.

**- PcComparator.Ressources :**

Ce projet contient notre page de styles, et quelques customControl. Il a une référence sur **PcComparator.Class** car il y a des TemplateSelector qui choisissent en fonction du composant.

**- PcComparator.Tests :**

Ce projet contient un test unitaire qui test toutes les méthodes de notre Manager. Il a donc des références sur **PcComparator.Class, PcComparator.Facade et PcComparator.Persistance** car un manager est créer avec un persistance et car nous manipulons de composants pour le tester.

**- PcComparator.WPF :**

Ce projet regroupe nos vues xaml. Il a des références sur **PcComparator.Class, PcComparator.Facade, PcComparator.Persistance et PcComparator.Ressources,** car ce projet instancie le manager. Nous n’avons pas pu enlever la référence sur **PcComparator.Class** car des vues sont obligées de se servir d’un attribut Composant pour pouvoir bind, ou bien cast sur un composant par exemple.

**-Setup :**

Ce projet contient le deploiement de notre application.Il n’a pas besoin de références.

Il semblait donc logique de découper notre projet de cette façon, car chaque sous-projet contient des « familles » de classes ou d’éléments bien différentes.