

Travaux Dirigés 8 :

Premiers Design Patterns

1 L'Usine

Un Design Pattern est une solution à un problème de programmation récurrent.

1.1 Principe

Un objet est créé en général par une instanciation simple de sa classe. Mais, dans un certain nombre de cas, il peut être utile de maîtriser sa création, par exemple, pour compter le nombre d'objets de ce type, ou pour adapter les paramètres.

Le Design Pattern correspondant s'appelle l'Usine (ou factory).

Ces caractéristiques sont :

nom : Usine (ou Factory).

contexte initial : besoin de contrôler la création d'objets.

problème : le constructeur d'une classe permet de contrôler comment un objet est instancié, par ne permet pas le contrôle de pré- ou post- conditions.

Forces en présence : Objet à créer, classe utilisant l'objet

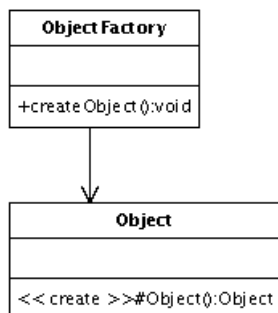
Solution : encapsuler la création dans une classe intermédiaire, qui contrôle les pré- ou post- conditions, ainsi que l'appel du constructeur. Pour que la solution soit mise en oeuvre systématique, le constructeur n'est plus public mais protégé (accès à l'intérieur du package).

Conséquence de l'application : accès direct au constructeur non possible.

Utilisations connues du pattern : limitation du nombre d'objets qui peuvent être créés.

Liens : Autres patterns de création.

1.2 Représentation



2 Sujet

L'objectif du TD est de créer un programme fonctionnel, par génération de code.

2.1 Scénario

Pour faire du pain, un boulanger a besoin d'un fournil : il ne peut pas travailler sans.

Le boulanger est modélisé par une classe 'Scenario.Boulangier'. Cette classe permet, dans la méthode main, de définir le comportement du boulanger.

Par conséquent, l'implémentation d'une boulangerie fait appel à une Usine, représentée par la classe 'Fournil'.

2.2 Le Fournil

Le fournil permet de fabriquer plusieurs types de pains :

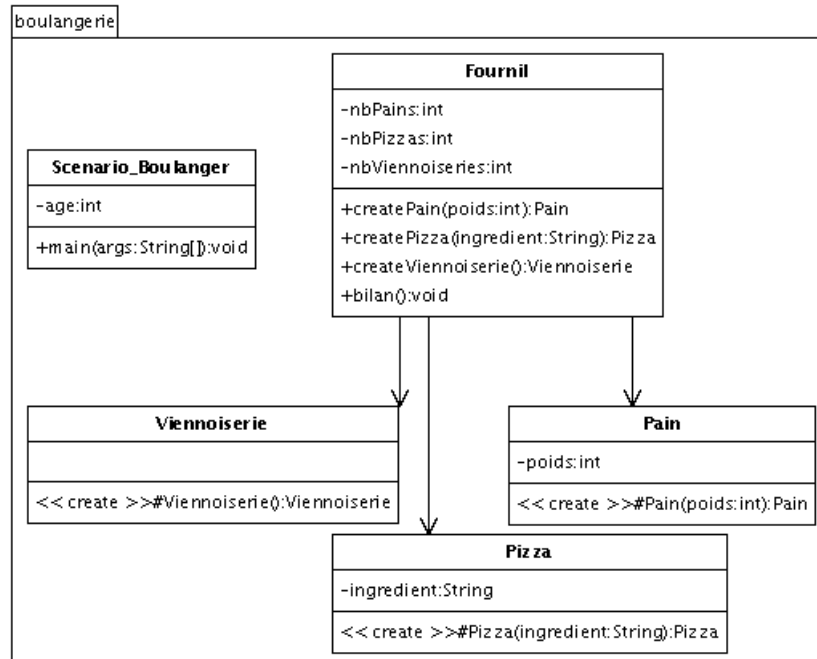
1. des pains, dont on peut faire varier le poids (paramètre de type 'int')
2. des Pizzas, dont on peut faire varier les ingrédients (paramètre de type String)
3. des Viennoiseries, simples (pas de paramètres).

Une méthode correspondante existe dans la classe Fournil. Les constructeurs des différentes classes (pains, pizzas, viennoiseries) sont protégés, et donc accessibles uniquement dans le package.

2.3 Contrôle de la production

Le Fournil permet de contrôler le nombre de produits fabriqués. Lors de la fabrication de chacun, une variable correspondante (nombre de pains, nombre de pizzas, nombre de viennoiseries) est mise à jour.

2.4 Représentation graphique



Created with Poseidon for UML Community Edition. Not for Commercial Use.