

Le penser objet

Alexis Nasr (d'après les slides de Arnaud Labourel)



Penser objet : décomposer le programme en objets

- Quels sont les objets nécessaires à la résolution du problème ?
⇒ décomposition du problème en **objets**
- À quels modèles des objets correspondent-il ?
⇒ définition des **classes**
- Quels sont les fonctionnalités/opérations dont on doit/veut pouvoir disposer sur ces objets ?
⇒ définition des **méthodes**
- Quelle est la structure des données de l'objet ?
⇒ définition des **attributs** des classes

Exemple de problème

- un catalogue regroupe des articles, il permet de trouver un article à partir de sa référence.
- un article est caractérisé par un prix et une référence que l'on peut obtenir. On veut aussi pouvoir déterminer si un article est plus cher qu'un autre
- une commande est créée pour un client et un catalogue donnés, on peut ajouter des articles à une commande, accéder à la liste des articles commandés ainsi qu'au prix total des articles et au montant des frais de port de la commande.
- un client peut créer une commande pour un catalogue et commander dans cette commande des articles à partir de leur références.

- un **catalogue** regroupe des **articles**, il permet de trouver un article à partir de sa **référence**.
- un **article** est caractérisé par un prix et une référence que l'on peut obtenir. On veut aussi pouvoir déterminer si un article est plus cher qu'un autre
- une **commande** est créée pour un **client** et un **catalogue** donnés, on peut ajouter des **articles** à une commande, accéder à la liste des articles commandés ainsi qu'au prix total des articles et au montant des frais de port de la commande.
- un **client** peut créer une **commande** pour un catalogue et commander dans cette commande des **articles** à partir de leur références.

- un catalogue regroupe des articles, il permet de **trouver** un article à partir de sa référence.
- un article est caractérisé par un prix et une référence que l'on **peut obtenir**. On veut aussi pouvoir **déterminer** si un article est plus cher qu'un autre
- une commande est créée pour un client et un catalogue donnés, on peut **ajouter** des articles à une commande, **accéder** à la liste des articles commandés ainsi qu'au prix total des articles et au montant des frais de port de la commande.
- un client peut **créer** une commande pour un catalogue et **commander** dans cette commande des articles à partir de leur références.

Description d'un catalogue

un catalogue regroupe des articles, il permet de **trouver** un article à partir de sa référence.

Méthodes :

- Item getItem(String reference)

Description d'un article

un article est caractérisé par un prix et une référence que l'on **peut obtenir**. On veut aussi pouvoir **déterminer** si un article est plus cher qu'un autre

Méthodes :

- `double getPrice()`
- `String getReference()`
- `boolean isMoreExpensiveThan(Item other)`

Description d'une commande

une commande est **créée** pour un client et un catalogue donnés, on peut **ajouter** des articles à une commande, **accéder** à la liste des articles commandés ainsi que prix total des articles et le montant des frais de port de la commande.

Méthodes et constructeurs :

- `Order(Client client, Catalog catalog)` (constructeur)
- `void addItem(Item item)`
- `List<Item> allItems()`
- `double getTotalPriceOfItems()`
- `double getShippingCost()`
- `Client getClient()`
- `Catalog getCatalog()`

Description d'un client

un client peut **créer** une commande pour un catalogue et **commander** dans cette commande des articles à partir de leur références.

Méthodes :

- `Order createOrder(Catalog catalog)`
- `void orderItem(Order order, String reference)`