

No de Travail pratique	Travail pratique 1
Étudiants	Wilfried NKOUEKAM MBOUGA, Nadine Manuela GOKAR (GOKN29569802),
Cours	MGL7361
Session	S2021
Enseignant	Naima Essadi

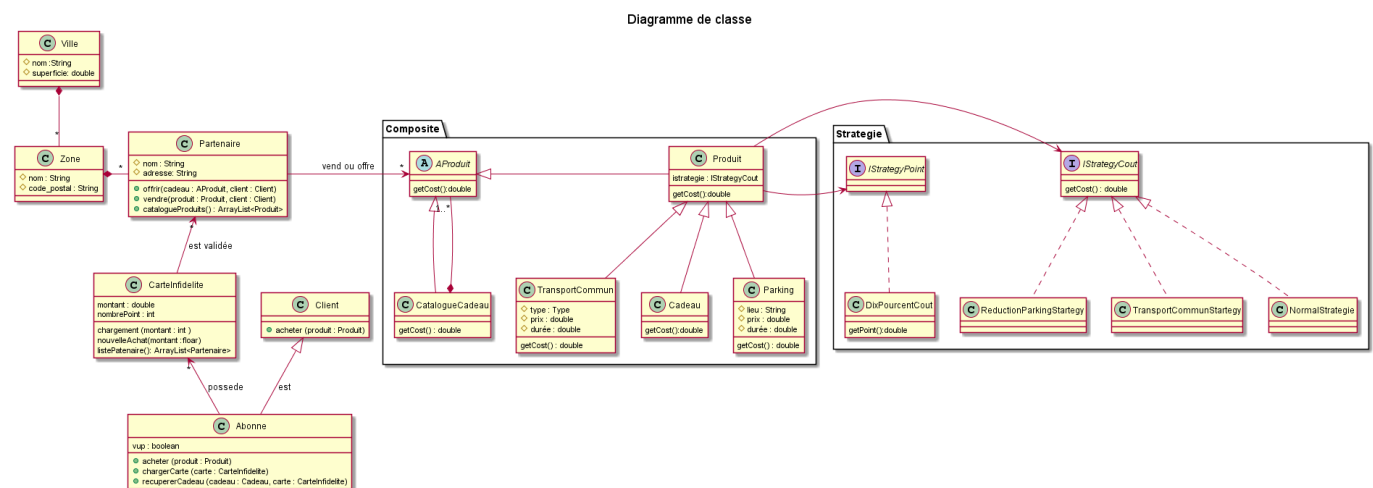
Justification des choix de conception

Type	Nom de l'objet	Responsabilités	Dépendances	Raison de dépendance	Raison choix Interface ou abstraction
Classe	Client	Identifier une personne qui désire acheter un produit			
Classe	CarteInfidelite	Identifier une carte d'infidelité	Partenaire	Identifier le nombre de partenaires ou la carte a été utilisé le mois précédent	
Classe	Abonne	Identifier un client qui dispose d'une carte d'infidelite	Client, CarteInfidelite	Identifier les informations clients ainsi que les informations de la carte d'infidelite détenues	
Classe	Ville	Identifier une ville et les zones associés	Zone	Identifier les zones de la ville	
Classe	Zone	Identifier une zone et les partenaires de la zone	Partenaire	Identifier les partenaires de la zone	
Classe	Partenaire	Identifier un magasin ou un commerce ainsi que les produits qui y sont vendues	AProduit	Identifier les produits et cadeaux vendues ou offerte par le partenaire	
Abstract class	AProduit	identifier une abstraction d'un service offert par un partenaire			Accéder facilement et de la même manière aux informations(descriptions,prix) d'un produit, d'un cadeau ou d'un catalogue cadeau

Type	Nom de l'objet	Responsabilités	Dépendances	Raison de dépendance	Raison choix Interface ou abstraction
Classe	Produit	identifier un service concret offert par un partenaire	Aproduit,IStrategyCout	Avoir les mêmes caractéristique et méthodes que l'on retrouve dans AProduit, identifier la méthode utilisée pour calculer le prix	
Classe	CatalogueCadeau	Identifier un ensemble de produit	AProduit	Un Catalogue de cadeau est composé de Produit abstrait et est aussi un produit abstrait	
Classe	Cadeau	Identifier un produit dont la méthode de calcul de prix est différente du la méthode normal	Produit	Un cadeau est un produit dont la stratégie de calcul de prix est différente de celle normale	
Interface	IStrategyCout	Identifier la strategie utilisée pour calcul le prix ou la reduction appliqué à un produit			Encapsuler à haut niveaux les variations possible qui pourrait avoir lieu lors du calcul de prix
Classe	DixPourCentCout	Effectuer une reduction de 10 pour cent sur le cout	IStrategyCout	correspond à une variante d'algorithme de calcul de cout de produit	
Classe	NormalStrategie	Identifier qu'aucune reduction n'est appliquée au produit	IStrategyCout	variante par défaut pour le calcul du prix d'un produit	

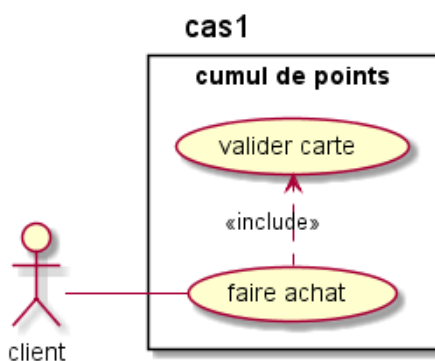
Type	Nom de l'objet	Responsabilités	Dépendances	Raison de dépendance	Raison choix Interface ou abstraction
Interface	IStrategyPoint	Identifier la strategie utilisée pour calcul le nombre de point appliqué à un produit			Encapsuler à haut niveaux les variations possible qui pourrait avoir lieu lors du calcul du nombre de points associe a un produit
Classe	DixPourCentPoint	Identifier que dix pour cent du cout du produit est considere comme nombre de point	IStrategyPoint	variante permettant de definir que dix pour cent du cout du produit est considere comme nombre de point	

Diagramme de classe



Réalisation des cas d'utilisation minimaux

Cas 1



Description détaillée du cas Nom: Effectuer un achat de produit

Objectif : Augmenter le nombre de points d'infidélité après achat dans une boutique

Acteur principal**Client:** Souhaite faire un achat et cumuler des points**Dates :** 2021-10-11**Version:** 0.1**Précondition**

Le client possède une carte d'infidelite

Scénario nominal

1. Le client fait un achat dans une boutique partenaire
2. Le client valide sa carte d'infidélité au passage en caisse

Le client peut revenir à l'etape 1 pour recommencer le processus

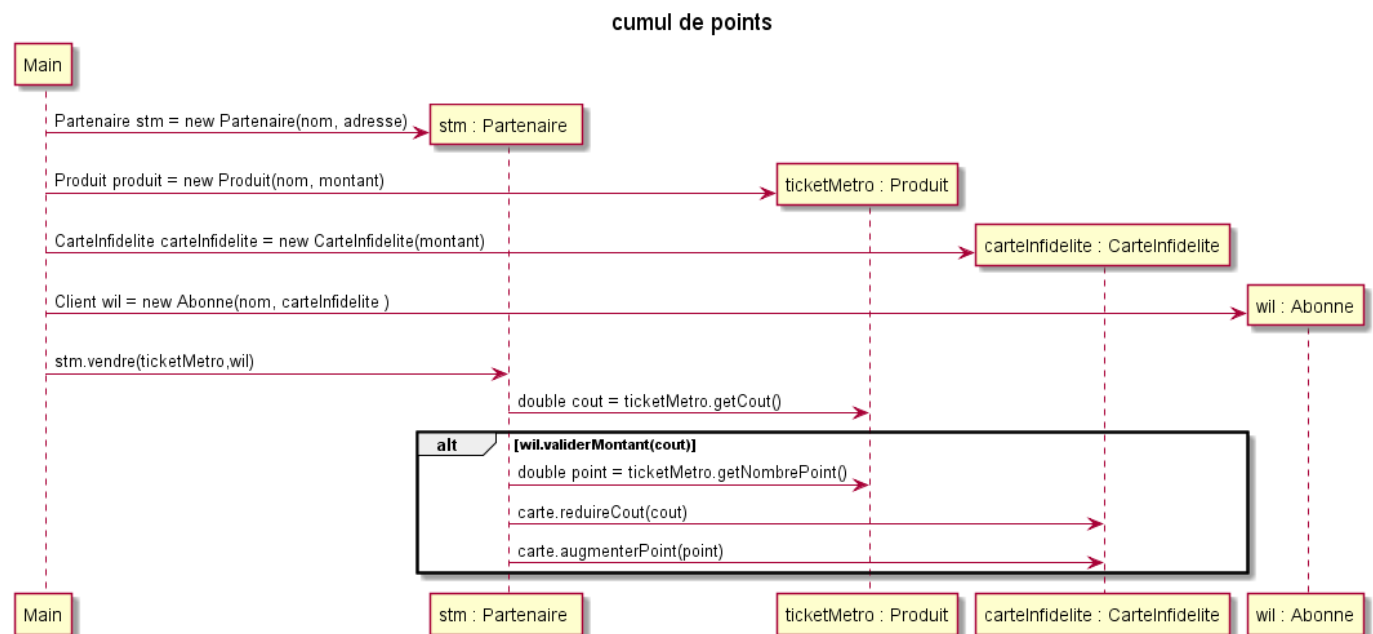
Post condition

1. Le système réduit le solde du client du montant de l'achat
2. Le système accumule 10% du montant d'achat en points sur la carte client
3. Le système ajoute le partenaire dans la liste des boutiques où le client a fait des achats pour ce mois

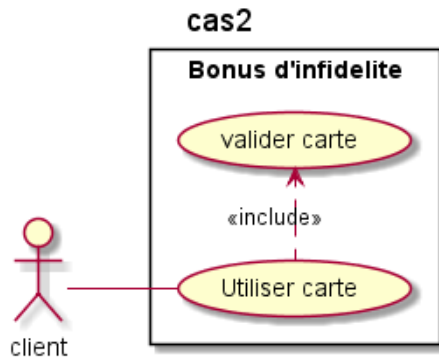
Extentions

- 2.La carte du client n'est pas validé
 - Le cas s'arrete
- 5.Le partenaire existe déjà dans la liste des boutiques où le client a fait des achats pour ce mois
 - Le cas s'arrête

Diagramme de séquence



Cas 2



Description détaillée du cas Nom: Bonus d'infidelite

Objectif : Augmenter le nombre de points d'infidelite à la fin de chaque mois

Acteur principal

Client: Obtenir bonus d'infidelite

Dates : 2021-10-11

Version: 0.1

Précondition

Le client possède une carte d'infidelite

Scénario nominal

1. Le client a utilisé sa carte dans N magasins pendant le mois qui se termine

Post condition

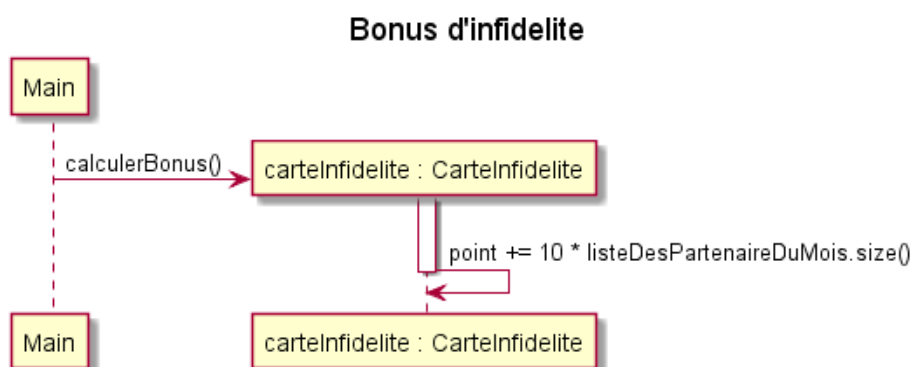
Le client reçoit un bonus de Nx10 points d'infidélité ($N > 1$)

Le client peut revenir à l'etape 1 pour recommencer le processus

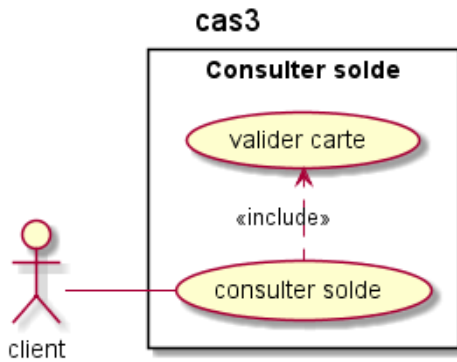
Extentions

- La carte du client n'a pas été utilisé
 - Le cas s'arrete

Diagramme de séquence



Cas 3



Description détaillée du cas

Nom: Consulter carte

Objectif : Consulter le solde de sa carte

Acteur principal

Client: Obtenir le solde de sa carte d'infidelite

Dates : 2021-10-11

Version: 0.1

Précondition Le client possède une carte d'infidelite\

Scénario nominal

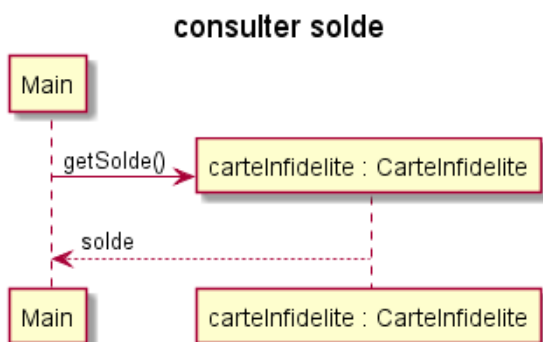
1. Le client choisit l'option de consulter son solde
2. Le système lui propose de valider sa carte
3. Le client valide sa carte
4. Le système lui affiche le solde de son compte

Le client peut revenir à l'etape 1 pour recommencer le processus

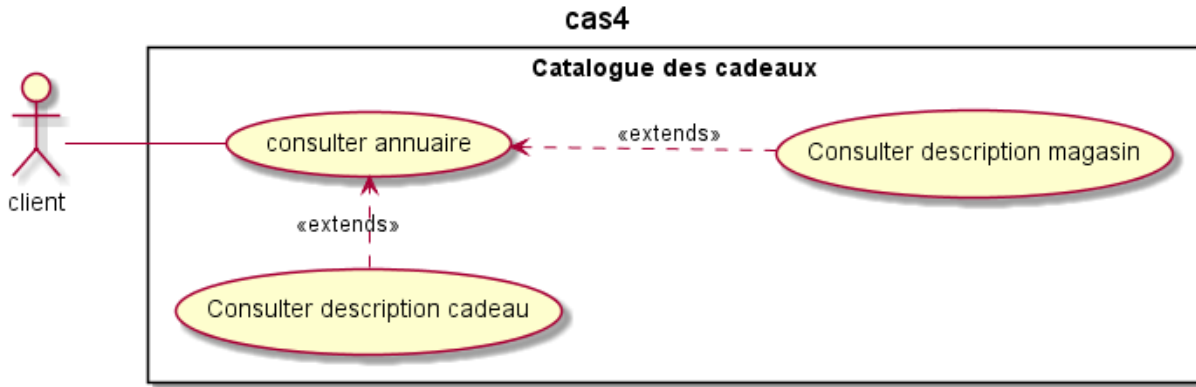
Extentions

- 3.La carte du client n'a pas été validé
 - Retour à 2

Diagramme de séquence



Cas 4



Description détaillée du cas

Nom: Catalogue cadeaux

Objectif : Consulter le catalogue de cadeaux

Acteur principal

Client: Obtenir les informations d'un catalogue de cadeaux

Dates : 2021-10-11

Version: 0.1

Précondition

Le catalogue existe

Chaque cadeau a un prix en points initialement définis

Scénario nominal

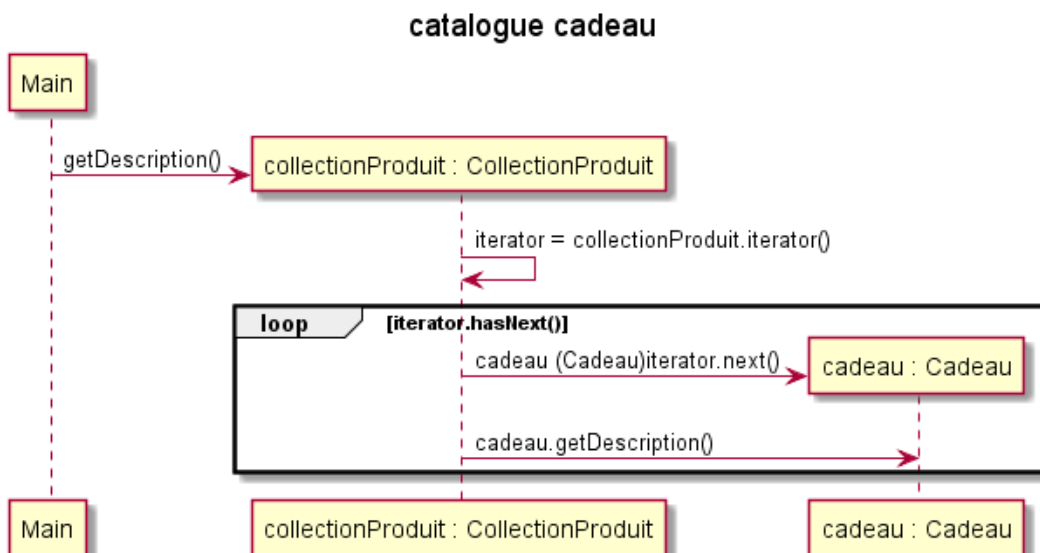
1. Le client choisit l'option de consulter le catalogue de cadeaux
2. Le système lui propose la l'annuaire des cadeaux disponibles

Le client peut revenir à l'etape 1 pour recommencer le processus

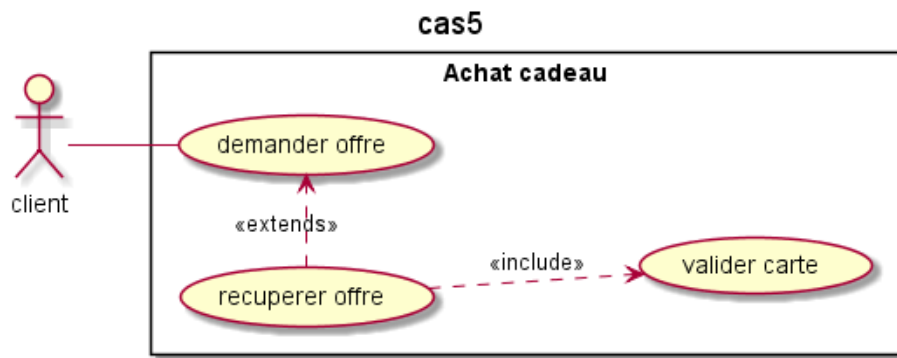
Extentions

- 3.Le client veut consulter le prix en point d'un cadeau
- 4.Le système lui affiche le cadeau et son prix en point
- 5.Le client veut consulter le magasin ou le produit est vendu
- 6.Le système lui affiche le magasin ou le cadeau est vendu

Diagramme de séquence



Cas 5



Description détaillée du cas Nom: Achat d'un cadeau en magasin

Objectif : vendre un cadeau a un client

Acteur principal

Client: Acheter un cadeau

Dates : 2021-10-11

Version: 0.1

Précondition

Le cadeau existe

Scénario nominal

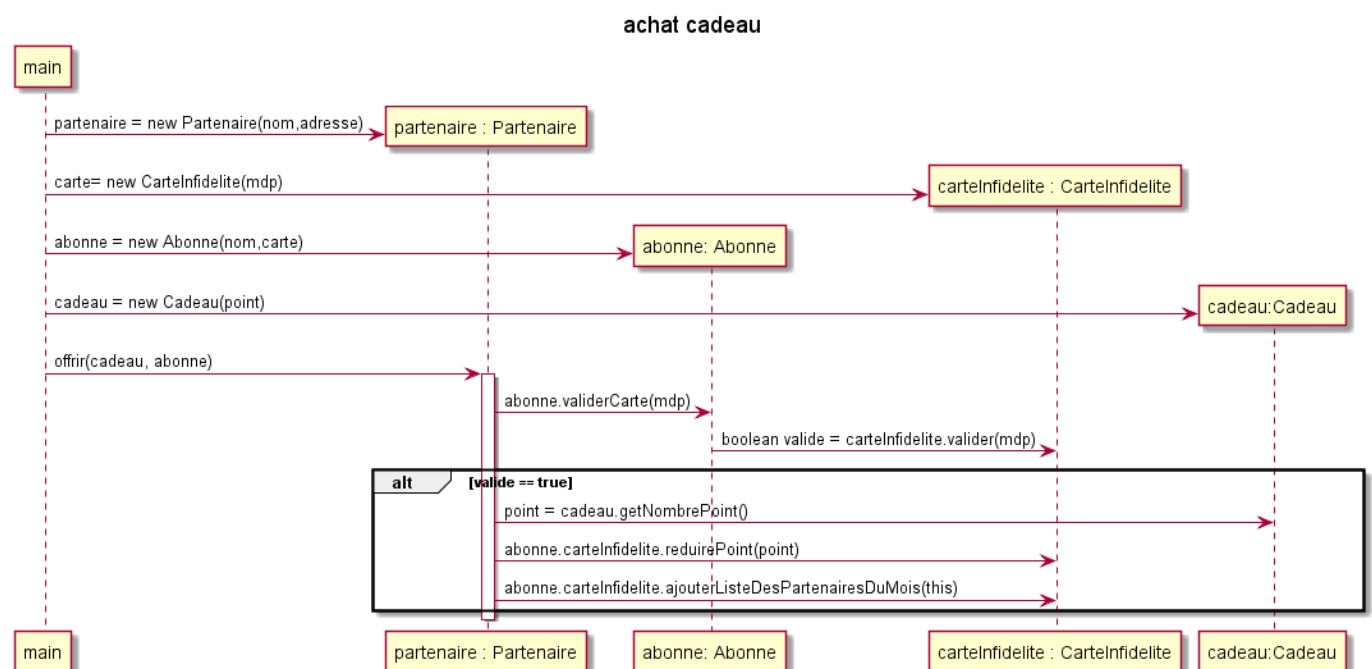
1. Il demande une des offres d'infidélité
2. Le magasin valide l'offre sur sa carte

Le client peut revenir à l'étape 1 pour recommencer le processus

post condition

Le solde de point du client est deduit du montant de l'achat

Diagramme de séquence



Justification de l'implementation

Patron composite

Nom dans le modèle de conception	Nom actuel
Primitive	AProduct
Composite	CatalogueCadeau
Leaf	Produit
method()	getNombreDePoints(), get Cout()

Nous nous sommes rendus compte qu'un catalogue cadeau ou catalogue de produit en réalité correspond à une collection de plusieurs produits sous un seul élément (Catalogue). Nous désirions facilement accéder à la description du catalogue de produit ou cadeau de la même manière que nous le faisons avec la description du produit. Sachant qu'à travers des itérations ou une boucle for nous pouvons parcourir le catalogue produit, nous avons choisis d'implémenté le catalogue et les produits sous forme de design pattern composite

Patron Strategie

Nom dans le modèle de conception	Nom actuel
Strategy	IStrategyCout
ConcreteStrategy	dixPourCentCout, NormalStrategie
doWork()	getCout():Double

Ici nous désirions protéger l'action de calculer les couts d'un produit des différentes variantes d'algorithme qui pourraient exister pour effectuer cette action. Nous avons choisi donc d'utiliser le design pattern Strategy