

Partiel de Programmation avancée (Groupe 1)

Enseignant : **Pierre Sylvain ILOGA BIYIK**, Durée : **3h**

PC et support de cours autorisés.

Le diagramme des classes UML de chaque pattern est donné en annexes (dernière page du sujet).

Exercice 1 : Le pattern Abstract Factory (8 pt)

Une entreprise de vente d'appareils électroniques gère des appareils *Android* et des appareils *Apple*. Cette entreprise met à la disposition de ses clients une page d'accueil en ligne dans laquelle ils peuvent consulter les caractéristiques des appareils disponibles. L'entreprise dispose de deux types d'appareils, il s'agit des *téléphones mobiles* et des *tablettes*. Chacun de ces appareils possède les caractéristiques suivantes dans la page d'accueil :

- Sa *marque* : chaîne de caractères.
- Sa *série* : entier positif.
- La *version de son système* : chaîne de caractères.
- Son *prix* (en euros) : entier positif.

Il vous est demandé de créer un nouveau projet *Java* composé d'un unique package nommé *pagedaccueil* qui contiendra aussi la fonction *main*. Votre programme va utiliser le pattern **Abstract Factory** pour créer le contenu de cette page d'accueil tel que décrit dans le Tableau 1 et commencer par afficher les informations sur le premier appareil. Ensuite, le programme doit permettre au client de parcourir ligne par ligne la page d'accueil à l'aide du clavier tel que décrit ci-dessous. Vous supposerez que le premier appareil n'a pas de précédent et que le dernier appareil n'a pas de suivant. En cas d'erreur lors de la saisie des données au clavier, le programme principal doit être relancé depuis le début.

'p': Appareil précédent
's': Appareil suivant
'q': Quitter la page
Votre choix : ____

Tableau 1 : Contenu de la page d'accueil

Type	Num	Marque	Série	Version	Prix	Famille
Téléphones mobiles	1	Iphone	8	10.2.4	900	Apple
	2	Iphone	6	12.1.0	700	Apple
	3	Motorola edge	20	11.4.3	300	Android
	4	Motorola edge	30	10.3.1	600	Android
	5	Iphone	11	13.1.3	1000	Apple
Tablettes	6	Galaxy Tab S	8	11.2.1	300	Android
	7	Galaxy Tab S	9	10.1.2	400	Android
	8	Ipad pro	10	12.2.4	1200	Apple
	9	Ipad pro	11	13.1.0	1500	Apple

Exercice 2 : Le pattern Builder (8 pt)

Une entreprise de vente d'appareils électroniques a développé un système de gestion des documents liés aux commandes de ses appareils. Ce système est manipulé par divers vendeurs travaillant pour l'entreprise. Lorsqu'un appareil est commandé, une *garantie d'achat* et une *facture* sont transmises au client par un vendeur au format *PDF* ou alors au format *HTML*. Chacun de ces documents contient le nom du client et l'identifiant unique de l'appareil commandé (entier entre 1 et 9 dans cet exercice). Seule la facture contient le prix en euros de l'appareil commandé (entier compris entre 300 et 1500 euros dans cet exercice). La garantie d'achat va aussi exclusivement contenir la durée en nombre de mois de la garantie (entier entre 3 et 24 dans cet exercice).

Il vous est demandé de créer un nouveau projet *Java* composé d'un unique package nommé *commande* qui contiendra aussi la fonction *main*. Votre programme qui va utiliser le pattern **Builder** pour implémenter le système de gestion de commandes de cette entreprise. Ce programme va demander au client de saisir au clavier les informations relatives à sa commande tel que décrit ci-dessous. Après cela, le programme va afficher à l'écran la liste des documents générés par le vendeur avec des messages adéquats. En cas d'erreur lors de la saisie des données au clavier, le programme principal doit être relancé depuis le début.

Nom client : ____

Identifiant appareil : ____

Prix appareil : ____

Durée de la garantie : ____

Format - (1) PDF (2) HTML : ____

Exercice 3 : Fusion des patterns Abstract Factory et Builder (4 pt)

Vous allez supposer dans cet exercice que les programmes *Java* rédigés aux exercices 1 et 2 sont destinés à la même entreprise de vente d'appareils électroniques. Le but de cet exercice est donc de fusionner les fonctionnalités de ces deux programmes en écrivant un unique programme permettant au client de parcourir la page d'accueil afin de commander un appareil parmi les appareils disponibles dans cette page. Pour y parvenir, vous allez commencer par supposer que la durée de la garantie de chaque appareil en nombre de mois est directement disponible dans la page d'accueil dont le contenu est désormais celui du Tableau 2.

Tableau 2 : Nouveau contenu de la page d'accueil

Type	Num	Marque	Série	Version	Prix	Garanti	Famille
Téléphones mobiles	1	Iphone	8	10.2.4	900	8	Apple
	2	Iphone	6	12.1.0	700	3	Apple
	3	Motorola edge	20	11.4.3	300	6	Android
	4	Motorola edge	30	10.3.1	600	10	Android
	5	Iphone	11	13.1.3	1000	15	Apple
Tablettes	6	Galaxy Tab S	8	11.2.1	300	8	Android
	7	Galaxy Tab S	9	10.1.2	400	12	Android
	8	Ipad pro	10	12.2.4	1200	18	Apple
	9	Ipad pro	11	13.1.0	1500	24	Apple

Vous allez aussi considérer que l'identifiant unique de chaque appareil est le numéro de la ligne contenant les informations sur cet appareil dans le Tableau 2. Enfin, les documents générés lors de la commande d'un appareil ne contiendront plus l'identifiant de l'appareil commandé, mais plutôt les informations sur cet appareil conformément au contenu du Tableau 2. Toutefois, seule la facture contiendra le prix de l'appareil et seule la garantie d'achat contiendra la durée de la garantie.

Il vous est demandé de créer un nouveau projet *Java* composé de trois packages nommés *pagedaccueil*, *commande* et *principal*. Le package *principal* est le seul qui contiendra la fonction *main*. Le package *pagedaccueil* va permettre au client de parcourir la page d'accueil comme dans l'exercice 1, avec en plus la possibilité pour le client de sélectionner l'appareil qu'il souhaite commander tel que décrit ci-dessous.

==== PAGE D'ACCUEIL =====

'p': Appareil précédent

's': Appareil suivant

'c': Commander un appareil

'q': Quitter la page

Votre choix : ____

Lorsque le client décide de commander un appareil, c'est le package *commande* qui prend la main et qui se charge de demander au client de fournir les informations relatives à sa commande tel que décrit ci-dessous afin de générer les documents relatifs à la commande de manière analogue à l'exercice 2.

==== COMMANDE =====

Nom client : ____

Identifiant appareil : ____

Format - (1) PDF (2) HTML : ____

Le package *principal* doit simplement faire appel aux autres packages pour permettre au client de parcourir la page d'accueil, puis de commander éventuellement un appareil de son choix. En cas d'erreur lors de la saisie des données au clavier, le programme principal doit être relancé depuis le début.

BONNE CHANCE !!!

Annexes : Diagrammes UML des patterns utilisés dans le sujet

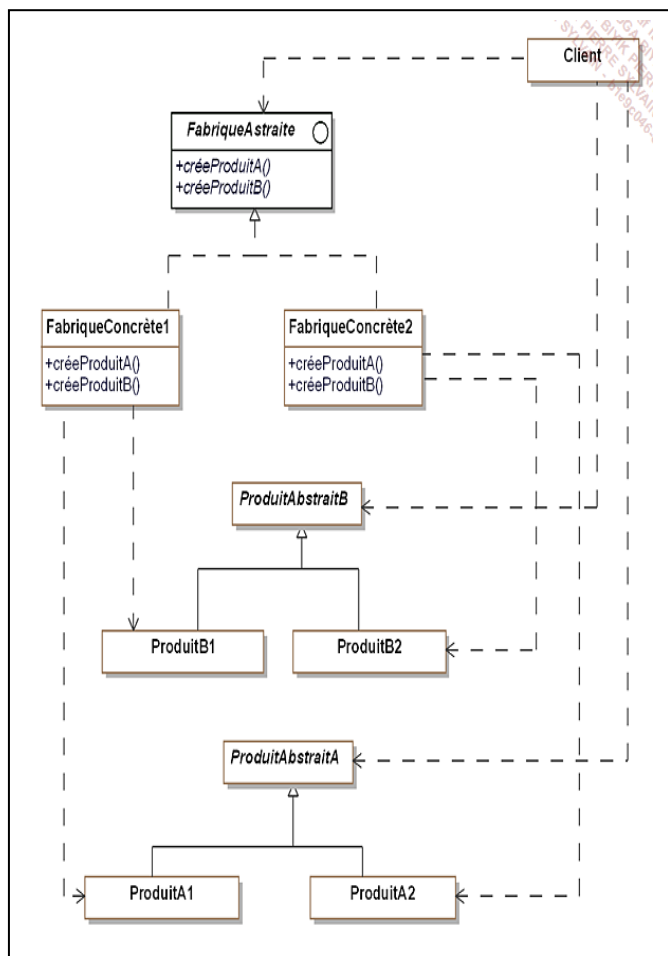


Figure 1 : Abstract Factory

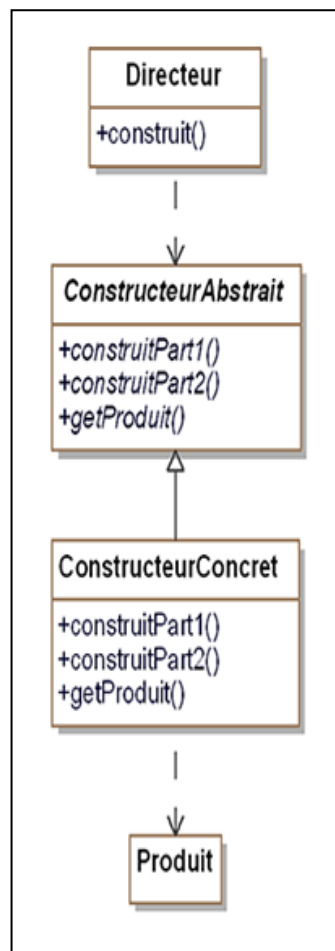


Figure 2 : Builder