Exercices pour application des patrons

Exercice 1

La compagnie Pro-Télé-Commande produit des télécommandes « universelles » pour contrôler des luminaires. La compagnie ne sait pas à l'avance quel genre de luminaire ses fournisseurs vont produire ni quels accessoires ces luminaires auront (exemple: un luminaire avec un ventilateur). Cependant, de façon générale, il est nécessaire d'allumer et éteindre la lumière et ses accessoires avec la télécommande.

Une télécommande produite par la compagnie Pro-Télécommande permet de contrôler un maximum de six (6) luminaires sans aucun accessoire (ou bien 3 luminaires avec ventilateur). Autrement dit, la télécommande universelle offre 6 boutons « On » et 6 boutons « Off ». La télécommande offre aussi six paires de boutons (« Dim » et « Inc ») pour ajuster l'intensité de la lumière dispensée par les luminaires lorsqu'ils sont allumés (Dim pour diminuer et Inc pour augmenter).

Vous devez proposer une conception qui permet de configurer la télécommande pour fonctionner avec les luminaires d'une maison donnée. Pour que votre conception soit flexible et qu'elle découple la télécommande des luminaires, vous devez appliquer le patron Commande vu en classe. Votre conception doit aussi contenir une classe de configuration qui permet de configurer le système pour une maison avec un luminaire avec ventilateur dans la cuisine, un luminaire dans une chambre et un luminaire dans le salon.

<u>Procédez progressivement comme suit pour aboutir à la solution</u>:

1) Identifiez les classes importantes à partir de l'énoncé.
2) Identifier les classes jouant les différents rôles définis dans la structure générique du patron Commande et faîtes votre diagramme de classes :
    a. Quelle(s) classe(s) joue le rôle de ConcreteCommand?
    b. Quelle(s) classe(s) joue le rôle de Invoker?
    c. Quelle(s) classe(s) joue le rôle de Receiver?
    d. Quelle(s) classe(s) joue le rôle de Client?

3) Utilisez des notes dans vos diagrammes de classes pour inclure le pseudocode des méthodes importantes de vos classes. Spécifiquement, vous devez illustrer le fonctionnement de votre télécommande en fournissant sous forme de pseudocode les méthodes pour allumer et éteindre un des luminaires et aussi pour réduire l'intensité de sa lumière.

Exercice 2

La compagnie BestPizza a ouvert plusieurs restaurants à Montréal, à Laval et ailleurs au Québec. Deux restaurants (par exemple : un de Montréal et un de Laval) offrent les mêmes catégories de pizzas à leurs clients (ex. pizza au fromage, pizza végétarienne, etc.). Cependant, chaque restaurant offre ses propres variations dans les ingrédients des pizzas.

La compagnie veut mettre en place une application qui lui permet de créer les pizzas appropriées selon le restaurant dans lequel la commande de pizza a été faite. Le choix d'une pizza se fait par son nom (méthode : *commander(nomPizza : String)*).

1) En utilisant un patron de conception vu en classe, proposez une conception pour produire des pizzas appropriées pour chaque restaurant de la compagnie. Limitez-vous à deux restaurants et un type de pizza dans votre conception. 2)
2) Écrivez le pseudocode des méthodes principales de votre conception.

Exercice 3

Faîtes le diagramme de séquence pour l'appel à la méthode *commander(nomPizza : String)* d'une classe représentant un restaurant dans la conception que vous avez proposée comme solution à l'exercice 2.

Exercice 4

Supposons que nous voulons construire un éditeur graphique simple qui permet de faire des dessins. L'éditeur offre un menu composé de plusieurs items correspondant à différentes actions. Parmi les actions qu'on peut exécuter sur un dessin, on peut créer des formes, supprimer des formes, copier ou couper une ou plusieurs formes, changer les propriétés d'une forme (couleur, taille, etc).

Nous voulons permettre à un utilisateur de défaire (undo) des actions et retourner le dessin à son état avant l'exécution de ces actions. Nous voulons aussi permettre à l'utilisateur de ré-exécuter (redo) une action qu'il vient de défaire. Tant que l'utilisateur n'a pas exécuté d'actions, il ne peut pas faire des « undo » ni « redo ». Quand l'utilisateur a exécuté une action, il peut faire un « undo » pour la défaire. Quand l'utilisateur a défait (undo) une action, il peut faire un « redo » pour la refaire.

Proposez une conception simple pour notre éditeur de formes et cela sous forme d'un diagramme de classes en utilisant du pseudocode pour illustrer le fonctionnement de votre conception. Votre conception peut combiner plusieurs patrons de conception. Vous devez justifier vos choix.

ex 1

Invoker : telecommande

Interface/classe abstraite Commande;

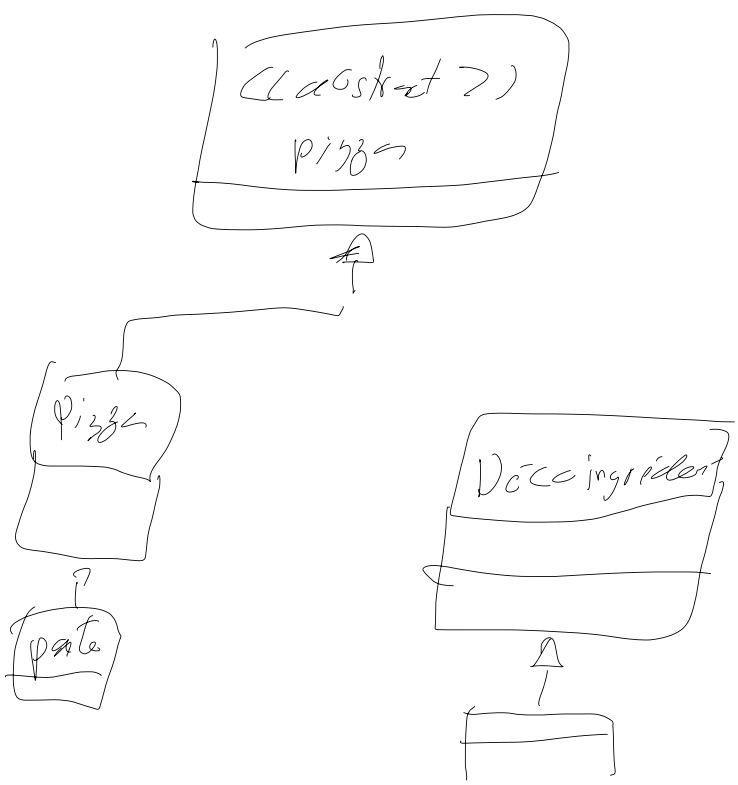
commandes concrete ; On, OFF, DIM, INC

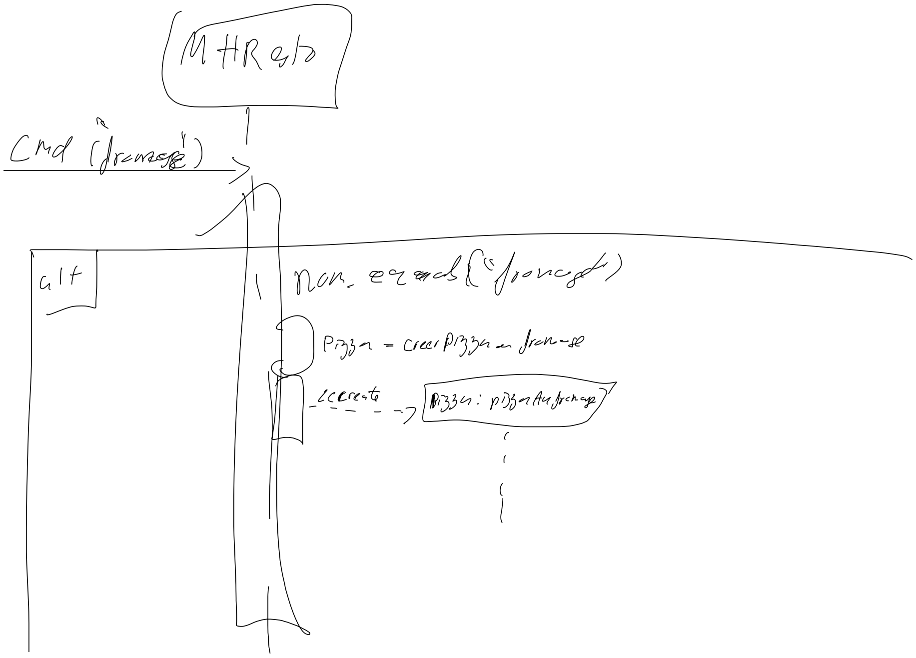
Receive ; luminaire/ventilateur

client : classe de config

ex 2

Commande (nom : string)





MHResto
Cmd ("fromage")
alt
nom.equals("fromage")
pizza = creerPizzaAuFromage
<<create>>
pizza : pizzaAuFromage