UML et POO – exercices

# Diagramme d’activité

## Exo 1 : commande

Le client passe commande au vendeur

Le vendeur demande au stock le contenu de la commande

Le vendeur livre le client

Le client règle la commande

À tout moment, le client peut annuler la commande

## Exo 2 : édition de rapports

Concevez un diagramme d’activité représentant l’édition de rapports de gestion de projets en tenant compte des détails suivants :

Le chef de projet saisit des données dans le système de gestion de projets.

Le système génère des rapports en vue de les imprimer.

L'impression d’un rapport n’a lieu que si le rapport est généré avec succès ; sinon, le système de gestion de projet génère une erreur.

Le système de gestion de projet peut exécuter d’autres traitements en même temps que la génération de rapports.

Lorsque le rapport est imprimé et les autres traitements terminés, le chef de projet peut choisir de générer un autre rapport.

## Exo 3 : processus de compilation et d’exécution de code managé .net

Décrire les 2 processus résumés ci-dessus par des diagrammes d’activité. Inutile d’utiliser des couloirs.

NB/ Ces processus sont décrits plus en détail sur [cette page MSDN](https://msdn.microsoft.com/fr-fr/library/k5532s8a(v=vs.110).aspx)

**Compilation**

Un compilateur compile les sources en MSIL (Microsoft Intermediate Language), et produit en parallèle des métadonnées qui décrivent le code.

Le MSIL et les métadonnées sont stockées dans un fichier portable exécutable

**Exécution par le runtime (CLR)**

Le compilateur JIT intégré au runtime convertit le code MSIL en code natif. Cette conversion est faite au moment de l’exécution, au fur et à mesure des besoins. Quand une méthode d’une classe est appelée, le compilateur JIT convertit son code MSIL en code natif et marque la méthode comme compilée afin que les appels suivants soient transmis directement au code natif.

Le code natif obtenu est stocké en mémoire pour qu’il soit disponible pour les appels ultérieurs

Au moment de la compilation en code natif, le code MSIL est soumis à un processus de vérification pour déterminer s’il est sécurisé, c’est à dire qu’il n’accède qu’aux emplacements mémoire autorisés. Si ce n’est pas le cas, une exception est levée.

# Diagramme de classes

## Exo 1 : l’hôtel

Un hôtel dispose d’un certain nombre de chambres. Chaque chambre dispose d’une salle d’eau avec douche ou baignoire. Un hôtel héberge des personnes, emploie du personnel et est dirigé par un directeur. On ne connaît que le nom et le prénom de toutes ces personnes.

Un hôtel a les caractéristiques suivantes : une adresse, un nombre de pièces et une catégorie (nombre d’étoiles). Une chambre est caractérisée par le nombre de lits qu’elle contient, son prix et son numéro. On veut pouvoir savoir qui occupe quelle chambre quelle date. Pour chaque jour de l’année, on veut pouvoir calculer le loyer de chaque chambre en fonction de son prix et de son occupation (le loyer est nul si la chambre est inoccupée). La somme de ces loyers permet de calculer le chiffre d’affaires de l’hôtel entre deux dates.

## Exo 2 : l’université

On a deux classes Université et personne.

Une université emploie plusieurs enseignants. Un enseignant peut être employé par plusieurs universités mais il n'y est pas obligé.

Une université a de nombreux étudiants qui pour leur part ne fréquentent qu'une seule université.

Une personne ne peut pas être à la fois étudiante et enseignante.

## Exo 3 : hiérarchie de systèmes de chauffage

Modéliser dans un diagramme de classes la hiérarchie de systèmes de chauffages décrite ci-dessous.

Préciser les éventuelles contraintes qui s’appliquent à certaines classes ou certains liens.

### Description

**Systèmes de chauffages**

Tous les chauffages ont des propriétés communes qui sont :

* La puissance maxi en Watt
* Le rendement énergétique
* L’intervalle entre maintenances
* La date de prochaine maintenance
* Le mode de fonctionnement courant
* Le réglage d’intensité

Un chauffage possède au minimum 2 modes de fonctionnement, qui sont Eteint et Allumé, mais le second peut être remplacé par plusieurs autres modes, selon le type de chauffage (ex : modes éco, hors gel…). On ne détaillera pas ici tous les modes de chaque chauffage.

Le réglage d’intensité s’exprime de façons différentes selon les types de chauffages. Il peut s’agir par exemple d’une température, d’un chiffre entre 1 et 10…etc. Mais cela reste une valeur numérique.

On peut réaliser les actions suivantes sur un chauffage :

* Sélectionner son mode de fonctionnement
* Régler son intensité
* Le réviser (faire sa maintenance) à une date donnée. Ceci permet entre autres de réinitialiser la date de prochaine révision à la valeur Date de la révision + intervalle révision.
* Calculer sa consommation d’énergie sur une période donnée

Hormis la première, ces actions sont réalisées de façons différentes selon les types de chauffages.

On distingue 3 types de chauffages :

* A combustion, comme les poêles à bois, chaudières à fuel…etc.
* Electriques
* A échange de chaleur, comme les pompes à chaleur, échangeurs géothermiques…etc.

**Un chauffage à combustion** peut fonctionner avec divers combustibles de la même catégorie, mais toujours un seul combustible à la fois.

Les catégories de combustibles sont : bois, fuel, gaz et granulés composites. On ne donne pas d’informations supplémentaires concernant ces catégories.

Un combustible est décrit par les caractéristiques suivantes :

* Un code
* Un libellé
* Une masse volumique
* Une capacité calorifique
* Une unité d’utilisation/stockage

Un combustible ne peut appartenir qu’à une et une seule catégorie.

La liste des unités pour les combustibles est fixe. Il y a le litre, le m3, la stère et le kg. On ne donne pas d’informations supplémentaires concernant ces unités.

Le chauffage à combustion nécessite un ou plusieurs tuyaux d’évacuation pour la fumée, caractérisés par leur diamètre et leur longueur.

Par ailleurs, certains chauffages à combustion nécessitent de vider régulièrement un bac à cendres.

**Le chauffage électrique** est constitué d’au moins une résistance chauffante caractérisée par sa puissance en watts et le métal dont elle est faite. La somme des puissances des résistances ne doit pas excéder la puissance maxi du chauffage global.

Le radiateur en lui-même doit être branché sur une installation électrique avec un fusible d’une intensité maxi adaptée.

Comme il n’y a aucune maintenance à réaliser sur ce type de chauffage, on peut considérer son intervalle entre maintenances et la date de prochaine maintenance comme infinis.

**Le chauffage à récupération de chaleur** fonctionne avec un fluide caractérisé par sa conductivité thermique et sa viscosité. Ce fluide est récupéré de l’extérieur et comprimé au moyen d’une pompe caractérisée par un débit maxi.

Le système de chauffage en lui-même est caractérisé par :

* la température minimale extérieure du fluide nécessaire à son fonctionnement. En effet, si l’air ou l’eau captés à l’extérieur sont trop froid, le système ne peut pas être mis en marche.
* la valeur de pression qu’il applique au fluide grâce à la pompe
* sa surface d’échange avec l’extérieur.

### Consignes

* Indiquer le caractère abstrait ou virtuel de chaque membre.  
  NB/ Les méthodes abstraites apparaissent en italique. On utilisera un stéréotype « virtual » pour mettre en évidence les membres virtuels.
* Pour rendre le diagramme plus lisible, ne pas reprendre les membres virtuels ou abstraits des classes ancêtres dans les classes dérivées, même s’ils devraient effectivement être redéfinis.
* Ne pas afficher de membres privés.
* Pour simplifier, on pourra utiliser des attributs publics pour modéliser les propriétés simples (ni abstraites, ni virtuelles), même si on utiliserait en réalité des propriétés dans le code C# (principe d’encapsulation).
* Renseigner les multiplicités sur les relations lorsqu’il est pertinent de le faire.

**Remarques pour moi**

Il n’y a pas qu’une seule solution :

* Des indices dans l’énoncé permettent de déterminer si les méthodes sont plutôt abstraites ou simplement virtuelles, mais on peut tout à fait les mettre toutes virtuelles, si on estime qu'on peut fournir une implémentation par défaut
* Chauffage électrique peut aussi être considérée comme abstraite

NB/ [cette page](http://www.climamaison.com/lexique/chauffage.htm) donne un lexique sur les installations de chauffage, qui peut être utile pour décrire ces dernières en termes d’objets

## Exo 4 : systèmes programmables

Certains chauffages sont programmables. Mais le caractère programmable peut s’appliquer à toutes sortes de systèmes, pas qu’aux chauffages.

Un programme est constitué d’un ensemble de plages de fonctionnement caractérisées par une heure de début, une heure de fin et un mode de fonctionnement

Il peut être activé ou désactivé.

On doit pouvoir définir ses différentes plages de fonctionnement, et vérifier que le programme est complet, c’est-à-dire qu’il couvre bien 24h.

On doit pouvoir également déterminer :

* Quel est le mode programmé à un instant donné
* Dans quelle plage de fonctionnement on se trouve à l’instant présent
* A quelle heure se produira le prochain changement de mode

Les plages de fonctionnement ne doivent pas pouvoir se chevaucher.

**A faire** : décrire ce système programmable à l’aide d’un diagramme de classes.

**Consignes** : le mode de fonctionnement peut être représenté par un simple entier.

# Diagramme de séquence

## Exo 1 : Mise en marche d’une chaudière

Il s’agit de décrire la séquence de démarrage d’une chaudière.

NB/ Cette description sert uniquement à des fins d’exercice, et ne prétend pas représenter la réalité.

Cette séquence fait intervenir les éléments suivants :

* La console électronique
* Le régulateur
* Le système d’alimentation en combustible
* La sonde de température

La console sert d’interface pour l’utilisateur ; elle permet d’interagir avec le régulateur, qui contrôle à la fois le système d’alimentation en combustible et la sonde de température

Lorsque l’utilisateur appui sur le bouton de mise en marche sur la console, ceci allume la console et lance une phase d’initialisation :

* La console demande au régulateur de s’initialiser.
* Celui-ci demande à son tour à l’alimentation en combustible, puis à la sonde de température de s’initialiser. Il obtient en retour de la sonde la valeur de la température courante, et la retransmet à la console pour affichage.
* La console affiche la température courante, ainsi que l’interface de saisie de la température de consigne. On considère alors la phase d’initialisation terminée.

L’utilisateur peut alors saisir une consigne de température, qui est retransmise au régulateur.

Le régulateur entre alors dans une phase récursive de régulation qui consiste à répéter les étapes suivantes tant que la température réelle est différente de la température de consigne :

* Si T° réelle < T° consigne, augmenter le débit de combustible
* Si T° réelle > T° consigne, diminuer le débit de combustible
* Attendre une minute et récupérer la valeur de la température réelle
* Transmettre cette valeur à la console pour informer l’utilisateur

NB/ Abréger le mot « température » en « T° » pour réduire la taille des libellés

Distinguer les messages asynchrones s’il y en a

## Exo 2 : paiement de l’addition au resto

Décrire dans un diagramme de séquence les étapes de paiement d’une addition à la fin d’un repas au restaurant.

On décrira les échanges entre le client, le serveur et l’appareil à carte bancaire, entre le moment où le client demande l’addition et le moment où il quitte le restaurant.

# Diagramme d’états-transitions

## Exo 1 : gestion de bugs

Le testeur, lorsqu’il trouve un bug dans l’application, ouvre un nouveau bug dans le système d’information.

Ce bug reste à l’état proposé jusqu’à ce que le développeur lise sa description et le passe à l’état actif pour indiquer qu’il va travailler dessus, ou bien à l’état « non reproduit » si la description n’est pas suffisante pour reproduire le bug.

Lorsque le bug est corrigé, le développeur le notifie au testeur en passant son état à « Corrigé »

Le testeur vérifie la correction. Si le bug est effectivement corrigé, il le clôture, sinon, il le repasse à l’état actif en indiquant la raison.

Il se peut que le testeur n’arrive pas lui-même à reproduire un bug qu’il avait proposé, auquel cas il le clôture