Exercices Diagrammes de classe

Exercice 1 (Poste)

Un timbre, une adresse et une enveloppe.

Une enveloppe prête à poster contient une adresse, et un timbre. Dans le cas contraire, elle n'est pas valide. Vu sous cet angle, la relation entre timbre, enveloppe et adresse est donc une composition.

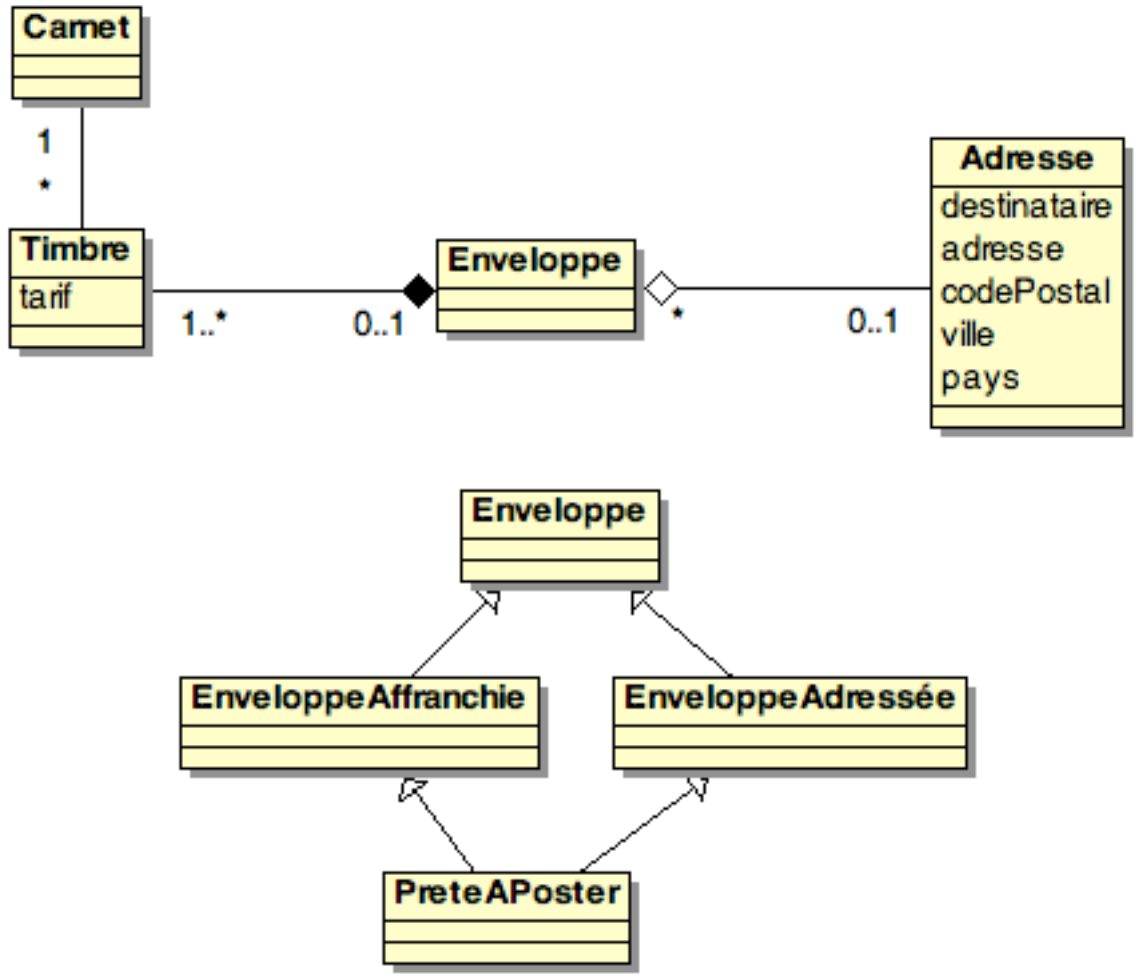
Si maintenant on considère que adresse est un concept à part entière, l'adresse peut être notée sur plusieurs enveloppes : elle est partagée, il s'agit d'une aggrégation.

Si on considère la vie des objets, le timbre est isolée ou en carnet, avant d'être collé sur l'enveloppe. Cette dernière est vierge avant utilisation.

Faire le diagramme de classes correspondant.

Mettre en évidence un héritage pour l'état de l'enveloppe (Affranchie, Adressée, PrêteAPoster) et sa liaison avec l'enveloppe.

Correction



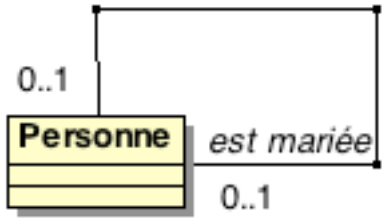
Exercice 2 (mariage, pacs)

Modélisez les différentes étapes de l'énoncé avec les contraintes associées :

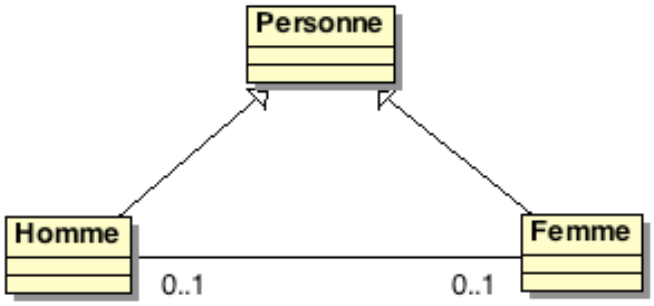
a) Deux personnes peuvent être mariées – suivant le droit français, une personne n'est pas tenue d'être mariée, mais ne peut pas être mariée avec plusieurs personnes à la fois.
b) Le mariage ne peut unit que des personnes de sexe opposé (Homme ou Femme)
c) Le PACS peut unit deux personnes (sans contraintes sur le sexe).
d) On ne peut pas être PACS2 avec soi-même, ni PACSE et marié en même temps.
e) Le mariage comme le PACS sont identifiés par une date, un lieu, et un contrat

Correction p122 de UML2 par la pratique

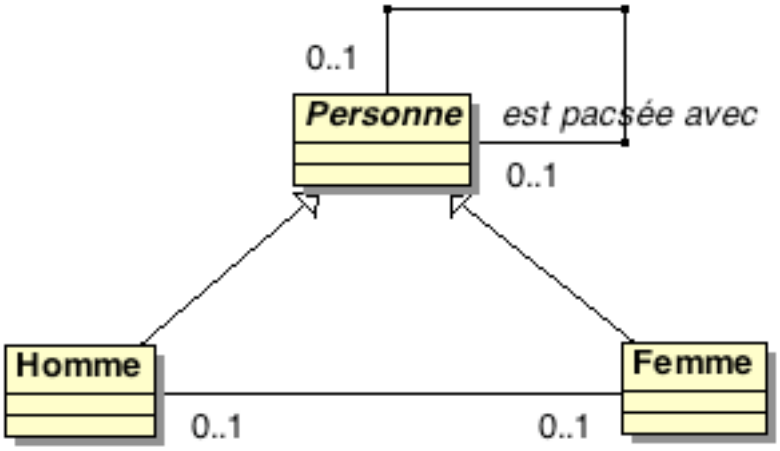
a)



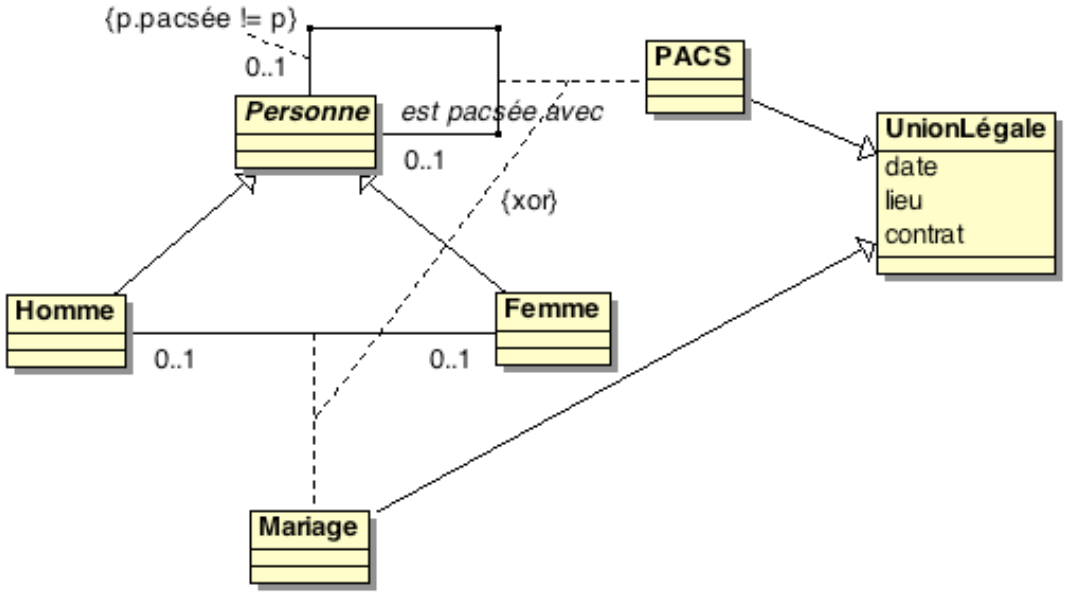
b)



c)

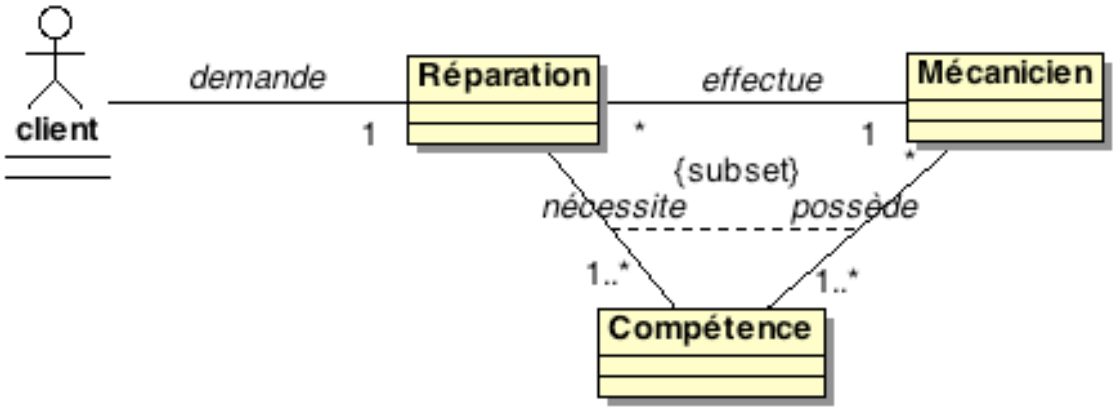


d)

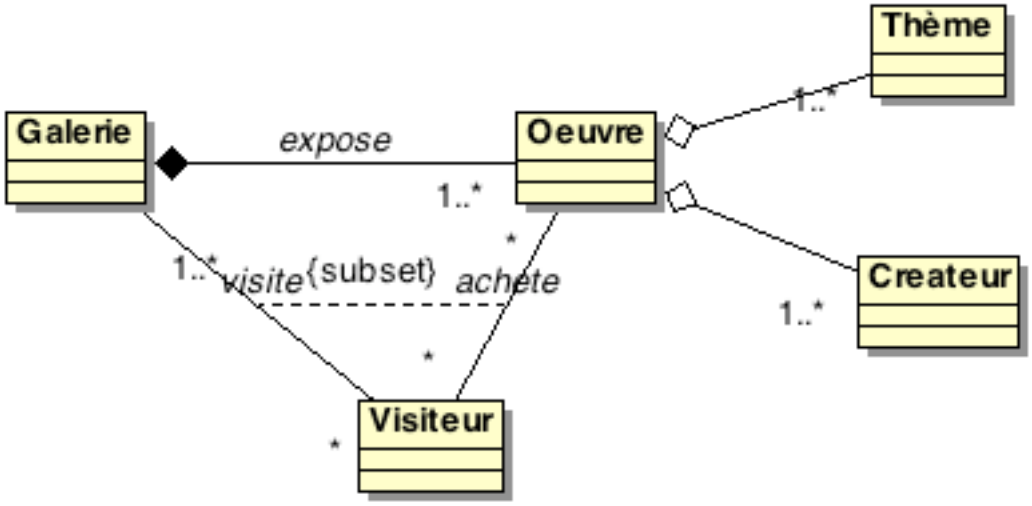


Exercice 3 (de Bruno Bouzy)

Dessiner les diagrammes (d'objets, de classes) correspondant aux situations suivantes :
(d) Un client demande une réparation. Une réparation est effectuée par un mécanicien ; Elle nécessite des compétences ; un mécanicien possède des compétences

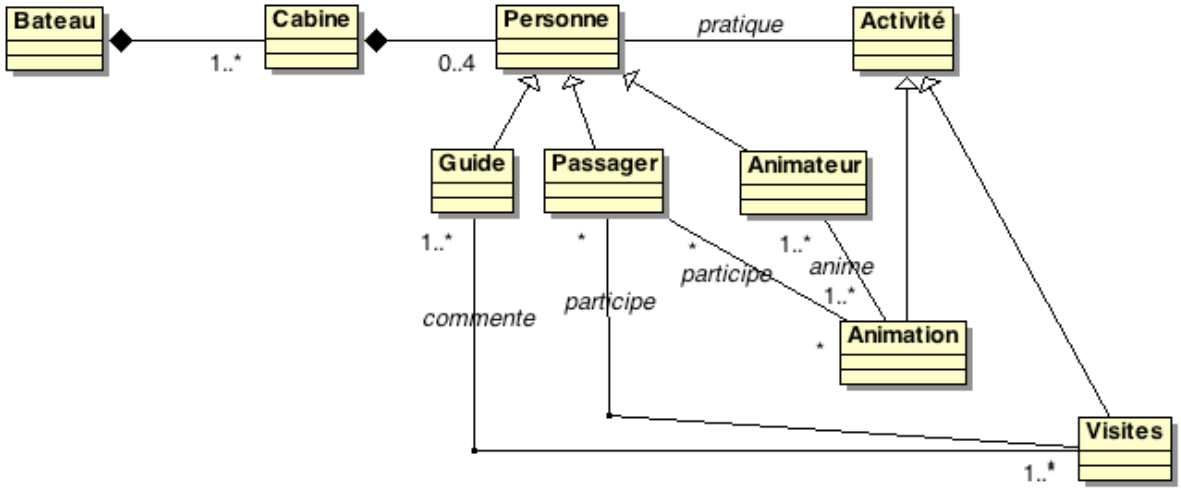


(e) Une galerie expose des oeuvres, faites par des créateurs et représentant des thèmes. Des clients, accueillis par la galerie, achètent des oeuvres.



les visiteurs qui achetent sont un sous-ensemble de ceux qui visitent la galerie

(f) un bateau contient des cabines, occupées par des personnes qui effectuent des activités. Les personnes sont ou bien des guides, ou bien des animateurs, ou bien des passagers. Les guides expliquent des visites aux passagers, et les animateurs animent des animations pour les passagers.



Note : dans cette correction, 4 personnes maximum occupent une cabine et une personne ne peut occuper qu'une cabine.

Exercice 4 (Brainstorming de Bruno Bouzy)

Préparer un diagramme d'objets montrant au moins 10 relations parmi les classes d'objet suivantes. Inclure les associations, agrégation, généralisations ; placer les ordres de multiplicité.

a) école, terrain de jeu, proviseur, conseil de classe, salle de classe, livre, élève, professeur, caféria, ordinateur, bureau, chaise, porte.
b) Chateau, douve, pont-levis, tour, fantôme, escalier, donjon, plancher, couloir, salle, fenêtre, pierre, seigneur, dame, cuisinier

c) Automobile, roue, frein, moteur, porte, batterie, silencieux, pot d'échappement

Correction faite en cours - pas de corrigé distribué

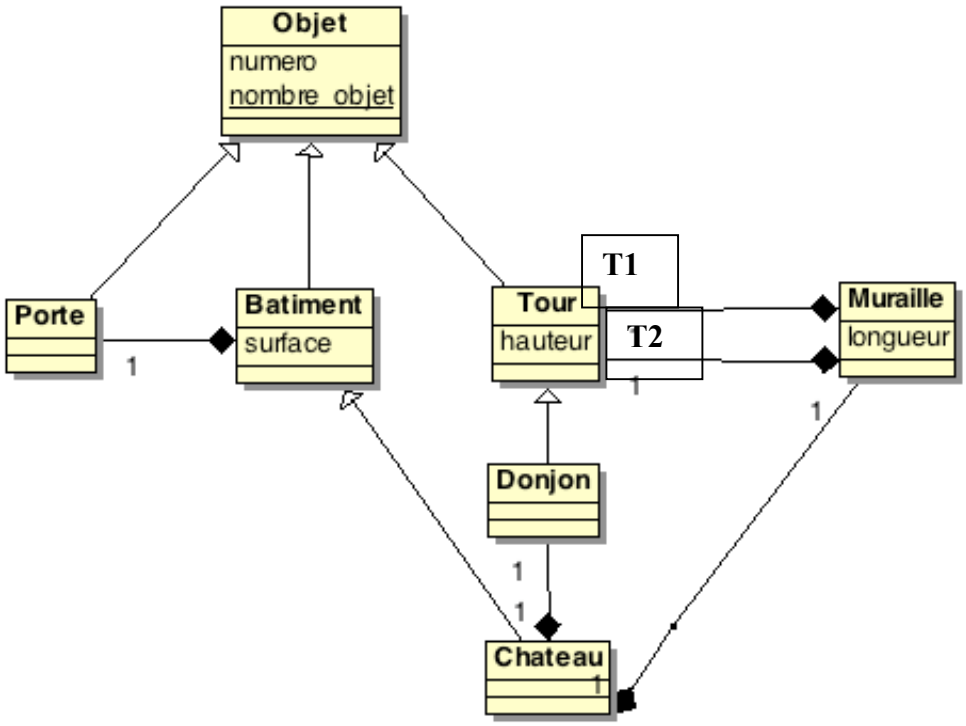
Exercice 5 (Reverse Engineering)

Rétro-ingénierie sur du code Java
http://java.sun.com/javase/6/docs/api/

Voici un ensemble de classes java – (voir l'annexe)

1) écrivez le diagramme Uml de classes correspondant
2) Trouvez ce qu'imprime l'exécution du main de Chateau

Corrections



2) Mon Chateau :

```
Le batiment :  objet numéro[1]
a pour surface[10000]
et une porte [La porte :
     objet numéro[2]]
La muraille :  objet numéro[3]
      a pour longueur[400]
      première Tour [La tour :  objet numéro[4]
            a pour hauteur[40]]
      seconde Tour [La tour :   objet numéro[5]
            a pour hauteur[60]]
Mon donjon :     [La tour :     objet numéro[6]
            a pour hauteur[80]]
```

Exercice 6 (Echiquier)

Le jeu d'échec se joue à 2, à tour de rôle, sur un échiquier carré, et chaque joueur possède initialement 8 pièces. Il ne peut bouger qu'une pièce par tour.

Un échiquier est un tableau de 8 par 8 cases, de couleur soit blanche (B), soit noire (N). Les rangées sont numérotées de 1 à 8, et les colonnes de a à h.

Il ne peut y avoir qu'une pièce maximum par case : c'est sa position.

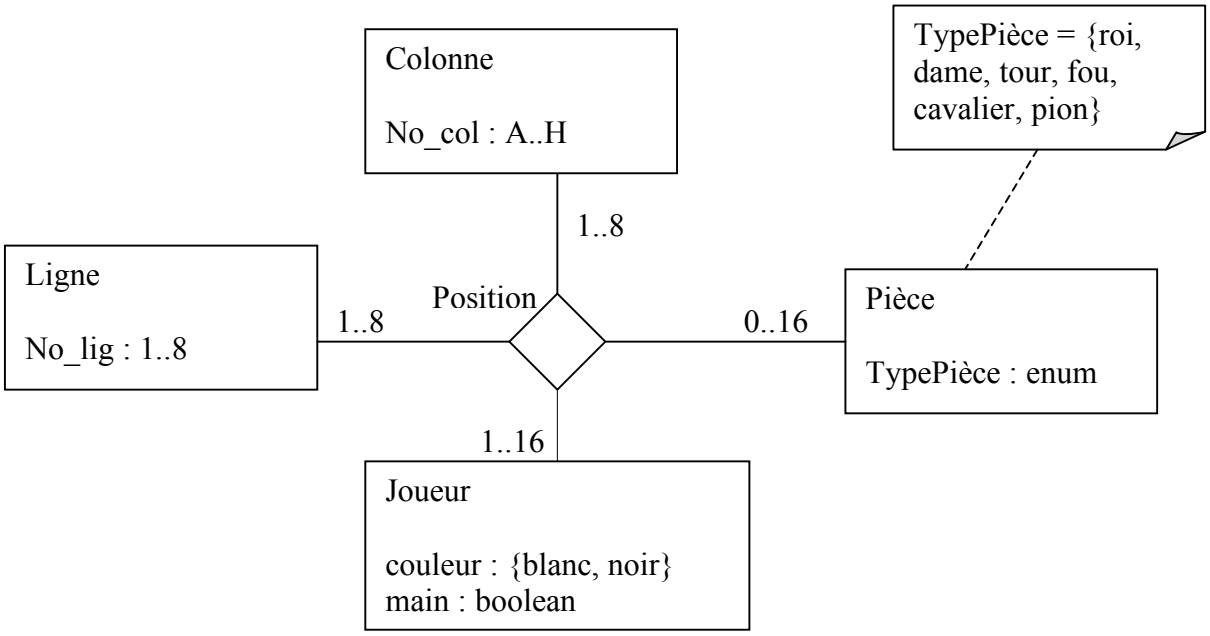
Les pièces sont de 2 couleurs (noir/blanc). Les figures sont roi (1), reine (1), fou (2), tour (2), cavalier (2), pions (8)

Un mouvement (ou coup) est caractérisé par la postion de départ et d'arrivée d'une pièce.

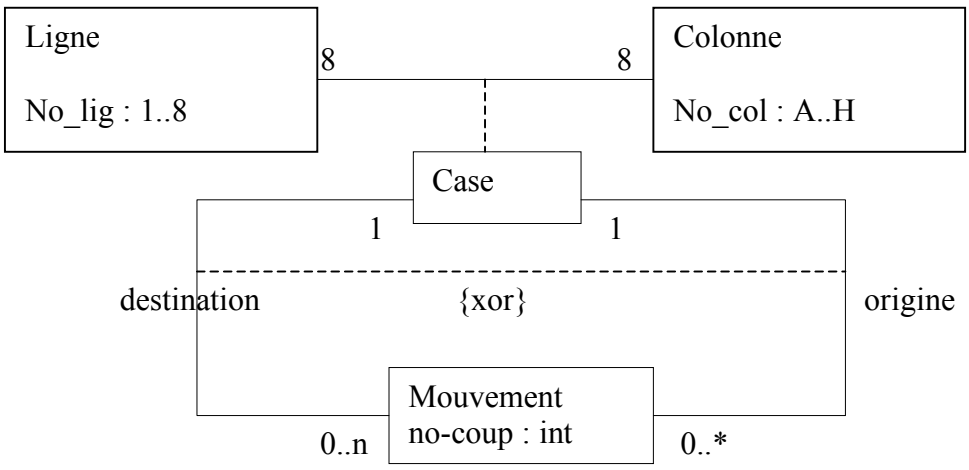
a) représenter une situation quelconque d'une partie d'échec en utilisant des classes-associations
b) Donner une modélisation pour la mémorisation des mouvements

Corrections p131 de UML2 par la pratique

a)



b)



Annexe :

```java
package TPUML;

public class Objet {
	static int nombre_objet = 0;
	int numero;

	public Objet(){
		nombre_objet++;
		numero = nombre_objet;
	}

	protected void finalize() throws Throwable {
		nombre_objet--;
	}

	public String toString(){
		StringBuffer buf = new StringBuffer("objet numéro["+numero+"]");
		return buf.toString();
	}
}
```

```java
package TPUML;

public class Porte extends Objet {

	public String toString(){
		StringBuffer buf = new StringBuffer("La porte : \n");
		buf.append("\t\t"+super.toString()+"");
		return buf.toString();
	}
}
```

```java
package TPUML;

public class Batiment extends Objet {
	Porte porte;
	int surface;

	public Batiment(int s){
		surface = s;
		porte = new Porte();
	}

	public String toString(){
		StringBuffer buf = new StringBuffer("Le batiment :");
		buf.append("\t"+super.toString()+"\n");
		buf.append("\t a pour surface["+surface+"] \n");
		buf.append("\t et une porte ["+porte.toString()+"] \n");
		return buf.toString();
	}
}
```

```java
package TPUML;

public class Tour extends Objet {

	int hauteur;
	Muraille maMuraille;

	public Tour(int h, Muraille m){
		hauteur = h;
		maMuraille = m;
	}

	public String toString(){
		StringBuffer buf = new StringBuffer("La tour :");
		buf.append("\t"+super.toString()+"\n");
		buf.append("\t\t\t a pour hauteur["+hauteur+"]");
		//buf.append("\t soutient la muraille["+maMuraille.toString()+"] \n");
		return buf.toString();
	}

}
```

```java
package TPUML;

public class Donjon extends Tour {
	Chateau monChateau;

	public Donjon(int hauteur, Chateau c){
		super(hauteur, null);
		monChateau = c;

	}

	public String toString() {
		StringBuffer buf = new StringBuffer("Mon donjon :");
		buf.append("\t ["+super.toString()+"]");
		return buf.toString();
	}
}
```

```java
package TPUML;

public class Muraille extends Objet {
	int longueur;
	Tour tour1;
	Tour tour2;
	Chateau monChateau;

	public Muraille (int longueur, int h1, int h2, Chateau c) {
		this.longueur = longueur;
		tour1 = new Tour(h1, this) ;
		tour2 = new Tour(h2, this) ;

	}

	public String toString(){
		StringBuffer buf = new StringBuffer("La muraille :");
		buf.append("\t"+super.toString()+"\n");
		buf.append("\t\t a pour longueur["+longueur+"] \n");
		buf.append("\t\t première Tour ["+tour1.toString()+"] \n");
		buf.append("\t\t seconde Tour ["+tour2.toString()+"] \n");
		//buf.append("\t appartient au chateau ["+monChateau.toString()+"] /n");
		return buf.toString();
```

```
        }
}
```

```
package TPUML;

public class Chateau extends Batiment {

                Donjon donjon;
                Muraille muraille;

                public Chateau (int longueur, int h1, int h2, int hd, int surface){
                        super (surface);
                        muraille = new Muraille (longueur, h1, h2, this);
                        donjon = new Donjon(hd, this);
                }

                public static void main(String[] args){
                        Chateau c = new Chateau(400, 40, 60, 80, 10000);
                        System.out.println(c.toString());
                }

                public String toString() {
                        StringBuffer buf = new StringBuffer("Mon Chateau : \n");
                        buf.append("\t "+super.toString());
                        buf.append("\t "+muraille.toString());
                        buf.append("\t "+donjon.toString());
                        return buf.toString();
                }
}
```