Génie logiciel
Licence informatique
Université de la Rochelle
Armelle Prigent

2019-2020

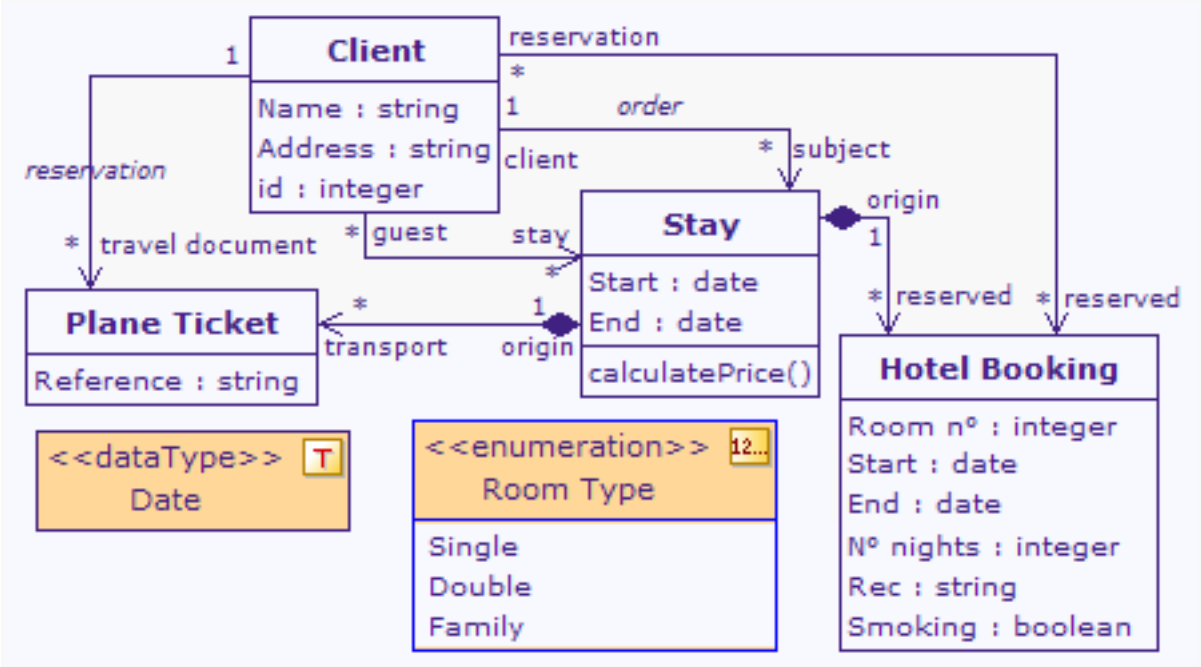
TP2
Modélisation objet avec UML :
Analyse des besoins et diagramme de classes

Evaluation des TPs - remise individuelle obligatoire
A la fin de la séance, vous déposerez sur moodle un compte rendu de TP avec les réponses aux questions posées et des captures écran des modélisations que vous avez réalisées. ***Le dépôt doit être fait en un seul fichier pdf et le nom du fichier doit être au format votre_nom_votre_prenom.pdf.***

Le code java écrit doit être intégré au rapport au format texte.
Les comptes rendus déposés sous forme d'archives ne seront pas évaluées

Exercice 1 : Préparer le développement à partir du diagramme de classes

A partir du diagramme de classes suivant, donnez les classes JAVA du système.



Dans ces classes JAVA, vous écrirez :
- La déclaration de la classe (avec l'héritage s'il y a lieu);
- Les variables d'instances et variables locales si nécessaire;
- Les entêtes de méthodes (**le code des méthodes n'est pas à développer**).

Génie logiciel
Licence informatique
Université de la Rochelle
Armelle Prigent

Exercice 2 : Reverse engineering

Sous Visual paradigm, vous modéliserez le diagramme de classes correspondant au code suivant disponible en annexe.

Exercice 3 : Conception d'un diagramme de classes

Donnez le diagramme de classes pour le système de gestion des inscription d'étudiants à des cours vu au TD/TP précédents.

Génie logiciel
Licence informatique
Université de la Rochelle
Armelle Prigent

2019-2020

Annexes

```java
package factory;

public class Main
{
 public static void main(String[] args)
 {
     Usine usine = new Usine();
     Unite unite = usine.formerUnite(TypeUnite.SOLDAT);
     Unite unite2 = usine.formerUnite(TypeUnite.MOTORISEE);

     System.out.println(unite);
     System.out.println(unite2);

 }

}
```

```java
package factory;

import java.util.ArrayList;

public class CommandantHumain extends Unite
{
public CommandantHumain()
{
   this.nom = "Lieutenant";
   this.coutConstruction = 14;
   this.precisionAttaque = 5;
   this.esquiveDefense = 2;
   this.equipements = new ArrayList();
}

public void equiper()
{
   this.equipements.add("Uzi");
   this.equipements.add("Bouclier");
   System.out.println("Equipement d'un commandant humain (Uzi, Bouclier).");
}
}

```

```java
package factory;

public class Usine
{
 private SimpleFabrique simpleFabrique;

 public Usine()
 {
     this.simpleFabrique = new SimpleFabrique();
 }

 public Unite formerUnite(TypeUnite type)
 {
     Unite unite = this.simpleFabrique.creerUnite(type);
     unite.consommerRessource();
     unite.equiper();
     return unite;
 }
}

```

```java
package factory;

import java.util.ArrayList;

public class UniteMotorisee extends Unite
{
public UniteMotorisee()
{
   this.nom = "Char";
   this.coutConstruction = 24;
   this.precisionAttaque = 12;
   this.esquiveDefense = 1;
   this.equipements = new ArrayList();
}

public void equiper()
{
   this.equipements.add("tourelle");
   System.out.println("Equipement d'une unite motorisee (Tourelle).");
}
}

```

```java
package factory;

import java.util.ArrayList;

public abstract class Unite
{
 protected String nom;
 protected int coutConstruction;
 protected int precisionAttaque;
 protected int esquiveDefense;
 protected ArrayList equipements;

 public void consommerRessource()
 {
     System.out.println("Consomme "+this.coutConstruction+" ressources pour la création de l'unité.");
 }

 public abstract void equiper();

 public String toString()
 {
     String str = "Nom : "+this.nom+"\n";
     str += "Coût de construction : "+this.coutConstruction+"\n";
     str += "Précision d'attaque : "+this.precisionAttaque+"\n";
     str += "Esquive en défense : "+this.esquiveDefense+"\n";
     str += "Equipements : ";
     for(int i=0; i<this.equipements.size(); i++)
     {
         str += this.equipements.get(i)+" ";
     }
     return str;
 }
}

```

```
1 package factory;
2
3 public enum TypeUnite
4 {
5 SOLDAT,
6 COMMANDANT,
7 MOTORISEE
8 }
9
```

```java
package factory;

import java.util.ArrayList;

public class SoldatHumain extends Unite
{
 public SoldatHumain()
 {
     this.nom = "Fantassin";
     this.coutConstruction = 5;
     this.precisionAttaque = 1;
     this.esquiveDefense = 2;
     this.equipements = new ArrayList();
 }

 public void equiper()
 {
     this.equipements.add("Pistoler");
     this.equipements.add("Bouclier");
     System.out.println("Equipement d'un soldat humain (Pistoler, Bouclier).");
 }
}
```

SimpleFabrique.java

```java
package factory;

public class SimpleFabrique
{
 public Unite creerUnite(TypeUnite type)
 {
     Unite unite = null;
     switch(type)
     {
         case SOLDAT:unite = new SoldatHumain();break;
         case COMMANDANT:unite = new CommandantHumain();break;
         case MOTORISEE:unite=new UniteMotorisee();break;
     }
     return unite;
 }
}
```