

## TP4 : Héritage et polymorphisme

### Exercice

L'hôtel « **ResortInfo** » a besoin du développement d'une nouvelle application desktop (en JAVA) pour la gestion des réservations effectuées par les clients qui souhaitent passer un séjour à Rabat. Cet hôtel propose à ses clients des chambres simples, des chambres doubles, et des suites royales.

- Les chambres simples se caractérisent par le numéro de chambre, boolean balcon et boolean télévision.
- Les chambres doubles se distinguent par le numéro de chambre, boolean balcon, boolean télévision et le nombre de lit.
- Les suites royales se caractérisent par le numéro de chambre, le nom de la suite, boolean balcon et boolean télévision.

### Règles de gestion :

- Les clients peuvent réserver une ou plusieurs chambres à l'hôtel mais une réservation ne concerne qu'une seule chambre à la fois (one-to-one). Une réservation est caractérisée par l'identifiant de la réservation, la date de début de réservation, le nombre de nuits et la chambre choisie et le nom du client.
- Notre programme se compose aussi d'une classe « **Test.java** » qui contient un **tableau** de type Réservation pour enregistrer les réservations des clients. Notre tableau est un attribut de la classe Test.java visible dans tout le projet. Proposition de la classe « **Test.java** » (voir l'annexe du tp4 sur Moodle).

```
package test;

import entity.Reservation;

public class Test {

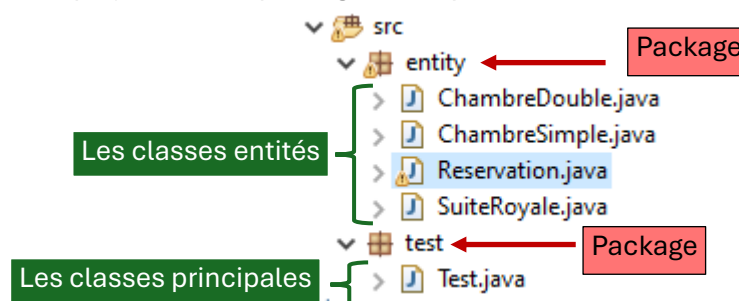
    public static Reservation[] tblResa = new Reservation[100];

    public static void main(String[] args) {

    }

}
```

- Structuré votre projet en deux package « **entity** » et « **test** » comme suit



## Questions

1. Proposer une structuration des classes (avec une relation d'héritage).
2. Donner le code JAVA pour l'implémentation des classes « **ChambreSimple.java** », « **ChambreDouble.java** », et « **SuiteRoyale.java** ». (Les classes, les attributs « **public** », les constructeurs, et toString).
3. Proposer une solution pour enregistrer les chambres d'hôtel dans la même structure de données.
4. Donner le code JAVA pour l'implémentation de la classe réservation « **Reservation.java** ».
5. Donner le code de la méthode Java qui permet d'ajouter une nouvelle réservation dans le tableau des réservations. (Les attributs « **public** », constructeurs, toString) (une proposition de cette fonction à compléter est partagé sur Moodle. (Voir l'annexe du tp4).
6. Donner le code JAVA d'une méthode qui permet de rechercher une réservation par son ID dans un tableau et qui renvoie sa position.
7. Donner le code JAVA de la méthode qui permet de trier le tableau des réservations en ordre croissant de la durée de chaque réservation.

**Tester vos méthodes dans le main de votre projet et déposer le compte rendu sur la plateforme Moodle.**