

# EXAMEN SESSION PRINCIPALE

Semestre : 1

Module : Architecture des Systèmes d'information II

Unité d'enseignement : Architecture N-tiers

Classe(s) : 4<sup>ème</sup> TWIN

Documents autorisés : OUI

Internet autorisés : NON

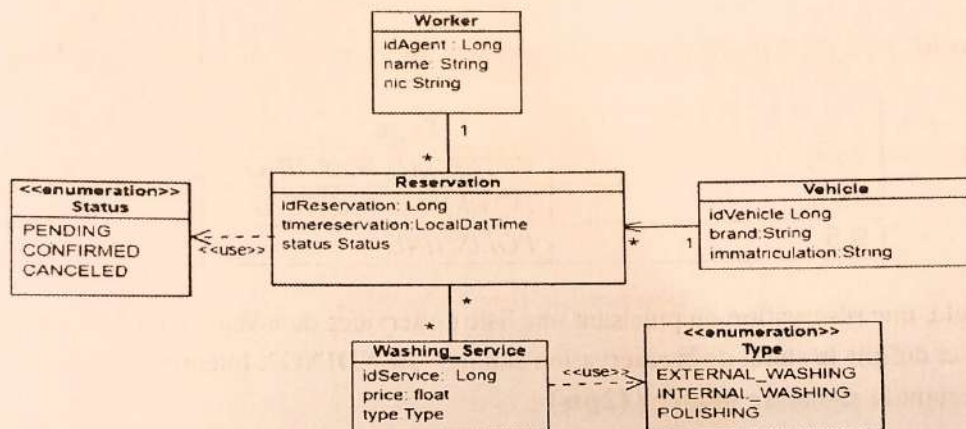
Date : 05/01/2024

Durée : 1H30 minutes

Nombre de pages : 3

**La validation de l'épreuve est appliquée sur la base d'un code source exécutable.  
Aucun code source non fonctionnel n'est comptabilisé lors de la validation.**

Vous êtes invité(e) à mettre en œuvre une application simplifiée pour la gestion d'une station de lavage.  
L'application présentera le diagramme de classe suivant :



## Partie I (6 points) :

Implémenter les entités qui permettent de générer le schéma de la base de données comme illustré dans le diagramme de classes, sachant que :

Les identifiants de toutes les entités sont auto-générés avec la stratégie « **IDENTITY** ».

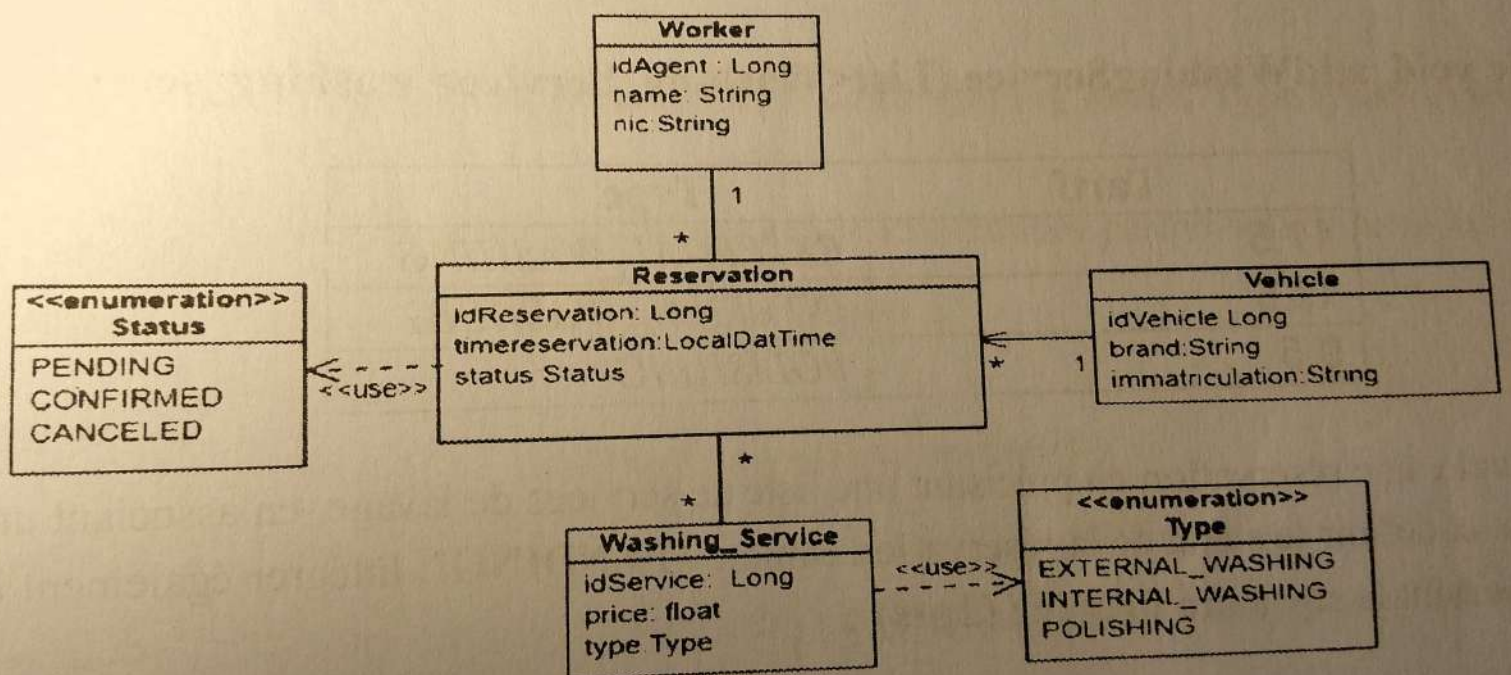
### Les relations bidirectionnelles :

- **Reservation – Washing\_Service** modélise l'affectation d'une liste des services à une réservation. (Washing\_Service est le fils).
- **Reservation - Worker** indique qu'une réservation est pour un seul worker et le worker peut avoir plusieurs réservations.

### La relation unidirectionnelle :

- **Vehicle - Reservation** indique qu'une réservation est pour un seul véhicule donné et le véhicule peut avoir plusieurs réservations

Les énumérations doivent être stockées en tant que chaînes de caractères dans la base de données (type d'énumération « **STRING** »).





## Partie II (14 points) :

Développer le code nécessaire dans une classe annotée par `@RestController` qui fait appel aux différents services.  
(Exposition des services avec Spring REST MVC).

Toutes les méthodes seront testées à travers **Swagger** ou bien **Postman**.

1. Ajouter 3 Véhicules ayant les détails ci-dessous en respectant la signature suivante (1pt) :

**public Worker addWorker (Worker worker)**

Name	nic
Kamila neutron	0098256
Sandra gozman	1583600

2. En utilisant la signature suivante, ajouter les trois services lavages en même temps ci-dessous (1pt) :

**public void addWashingService (List<Washing\_Service> washing\_service)**

Tarif	Type
17.5	EXTERNAL_WASHING
13.5	INTERNAL_WASHING
9.5	POLISHING

3. Ajouter un véhicule, une réservation en précisant une liste de services de lavage, en associant un véhicule à cette réservation, et définir le statut de la réservation comme "**PENDING**". Intégrer également les détails suivants, en respectant la signature fournie : (2pts) :

**public Vehicule addVehicleReservationAndAffectToWashingservice (Vehicule vehicle, List<Long> idService);**

- **Reservation 1:**  
Vehicle: BMW 1006TN04  
Washing Service: {EXTERNAL\_WASHING, POLISHING}
- **Reservation 2:**  
Vehicle: Audi 1087TN08  
Washing Service {INTERNAL\_WASHING}
- **Reservation 3:**  
Vehicle: Jeep 2106TN13  
Washing Service {EXTERNAL\_WASHING, INTERNAL\_WASHING}
- **Reservation 4:**  
Vehicle: Audi 1087TN08  
Washing Service {EXTERNAL\_WASHING, POLISHING, INTERNAL\_WASHING}

**NB :** La date est stockée avec l'année, le mois, le jour, l'heure, la minute et la seconde

**(TIMESTAMP)**

Date heure réservation = Date heure système + 2 heures



4. Créer un aspect permettant d'afficher le message « **Waiting for a Worker** » après le bon déroulement de la méthode **addVehicleReservationAndAffecttoWashingService** (1,5 pts).

5. Attribuez un worker à une ou plusieurs réservations en définissant le statut de la réservation comme "CONFIRMED", tout en veillant à ce que le nombre maximal de réservations pour un worker soit égal ou inférieur à 5, en respectant la signature suivante. (2pts).

```
public Worker affectWorkertoReservation (String nic, List<Long> idReservation)
```

- Kamilia Neuton → idsReservation [1 2 4]
- Sandra Gozman → idsReservation [3]

6. Concevoir un service permettant de calculer le revenu total d'un worker pour un service de lavage spécifique, en tenant compte uniquement des réservations « **CONFIRMED** » sur ce service pour le mois courant (2pts).

```
public Float calculateTotalIncomePerWorker (String name, Type typeService)
```

7. En utilisant **Spring Scheduler**, créez un service toutes les 15 secondes permettant de lister les réservations dont son statut est **PENDENIG**, triées en fonction de leur date.(2pts) :

```
public void getReservation() ;
```

8. Retourne la liste des services de lavage accompagnée du nombre de réservations associé à chaque service, en respectant la signature suivante (2.5pts):

```
public Map<String, Long> getListServiceAndNbreservation() ;
```

Format:

```
String: Type Washing_Services    Long: Nbr de reservation
```

Exemple:

```
{  
    "POLISSAGE": 2,  
    "EXTERNAL_WASHING": 4,  
    "INTERNAL_WASHING": 1  
}
```

Bon Courage

