



Contrôle (Module : Java)
(1h30min)

Exercice 1 :

Un centre commercial au Maroc souhaite gérer son parking de véhicules (un ensemble d'emplacements pour stationner les véhicules) via une application Java simple. Le parking est géré par deux responsables : le responsable d'entrée au parking indique le numéro de l'emplacement libre pour le stationnement du véhicule et le responsable de sortie du parking calcule le montant à payer par le conducteur selon la durée de stationnement du véhicule. L'application contiendra un ensemble de classes, à savoir : **Vehicule**, **Horaire**, **Emplacement** et **Parking**.

On considère les déclarations suivantes :

```
public class Vehicule{
    private String mat; // Le matricule du véhicule
    private String marq; // La marque du véhicule
    private String typ; // Le type du véhicule
    Vehicule(String mt,String mr,String tp) {mat=mt;marq=mr;type=tp; }
    public String toString()
    {String ch="Matricule: "+mat+"Marque: "+mar+" Type: "+ typ; return(ch);}
    String getType(){return typ;}
    String getMarq(){return marq;}
    String getMat(){return mat;}
}

public class Horaire{
    private int heur; // Le nombre d'heures
    private int minu; // Le nombre de minutes
    Horaire(int h,int m) {heur=h ;minu=m ;}
    String toString()
    {String ch=" "+heur+" heures "+minu+" Minutes" ;}
    int getHeur() {return heur ;}
    int getMinu() {return minu ;}
}
```

1) Ecrire la classe Emplacement :

(8pts)

- o Numéro de l'emplacement (**num** : int)
- o Etat de l'emplacement (**eta** : boolean) // true pour libre et false pour occupé
- o Véhicule stationné (**veh** : Vehicule)
- o Heure de début de stationnement (**deb** : Horaire)
- o Un constructeur **Emplacement** pour initialiser le numéro et l'état libre (par true)
- o Une méthode **clavier** qui permet de saisir et retourner une chaîne de caractères
- o Une méthode **stationner()** pour saisir les informations concernant le stationnement d'un véhicule.
- o Une méthode **Horaire** **dureeStationnement(Horaire fin)** qui calcule et retourne la durée de stationnement par rapport à l'heure de quitter le parking.
- o Une méthode **int fraisStationnement(Horaire fin)** qui calcule et retourne les frais de stationnement en utilisant le tableau ci-dessous.

Type de véhicule	Durée <= 1h	1h < Durée <= 2h	Durée > 2h
Moto	10 dh	5 dh	2 dh
Voiture	15 dh	10 dh	5 dh



(Exemple : pour une moto : si la durée = 2h30min alors les frais : $10 + 5 + 2 = 17$ dh)

- o La méthode `String toString()` qui retourne une chaîne de caractères contenant toutes les informations.

2) Ecrire une classe `Parking` répondant au cahier des charges suivant :

(9pts)

- o Un tableau d'emplacements (`park[]` : Emplacement)
- o Le nombre maximum d'emplacements (`max` : int)
- o Le nombre d'emplacements dont l'état est libre (`nl` : int)
- o Les recettes des emplacements (`recettes[]` : int)
- o Un constructeur `Parking` pour initialiser le nombre maximum des emplacements et le nombre des emplacements libres.
- o Une méthode `int rechercheLibre()` qui fait une recherche à partir du début et retourne le numéro du premier emplacement libre. La fonction retourne -1 dans le cas où le parking est plein.
- o Une méthode `void accederParking(Vehicule v, Horaire acc)` qui simule le stationnement d'un véhicule dans le parking (c-à-d ajouter un véhicule au tableau « park »).
- o Une méthode `void quitterParking(Vehicule v, Horaire quit)` qui simule la sortie d'un véhicule du parking.
- o Une méthode `void afficher()` qui permet d'afficher la liste des véhicules stationnés dans le parking.
- o Une méthode `int calculRecettes()` pour calculer et retourner la recette totale reçue à la fin de la journée.

3) Réaliser une application JAVA pour tester la classe `Parking`

(3pts)