

SAE 101 : La gestion d'un parc de voitures

Objectifs :

- Implementation du paquetage *voiture*
 - Travailler les enregistrements et les sous-programmes
-

Au cours de ce SAE vous allez développer une application en Ada pour la gestion d'un parc de voitures. Pour chaque véhicule, il est demandé de renseigner le numéro d'immatriculation, le numéro de place du parking et l'heure d'arrivée (en secondes).

Dans un premier temps, nous aimerons programmer le paquetage *voiture* qui comporte :

- La définition du type voiture nommé *Type_Voiture*
- Un ensemble de sous-programmes permettant la manipulation du *Type_Voiture*.

Ainsi, vous allez construire les jeux de tests fonctionnels pour s'assurer que votre code satisfait le comportement attendu.

En ce qui concerne le dépôt de cette semaine : déposez les sources de votre projet Ada dans un dépôt Moodle à 18h30.

1 Modélisation du type *Type_Voiture*

Quelle sont les informations à stocker pour chaque voiture ? Quel est le type de donnée pour chaque information ? Proposer une structure de données complexe permettant de regrouper les plusieurs données d'une seule voiture.

2 Programmer le paquetage *voiture*

1. Créez les deux fichiers du paquetage *voiture* (*voiture.ads* et *voiture.adb*).

Piqûre de rappel :

- La spécification du paquetage contient les définitions des types et les en-têtes des sous-programmes (fichier avec l'extension .ads)
- Le corps du paquetage contient les corps des sous-programmes (fichier avec l'extension .adb)

2. Définissez le type de donnée *Type_Voiture* qui stocke le numéro d'immatriculation *num*, le numéro de place du parking *place* et l'heure d'arrivée *heureEntree*. Le numéro d'immatriculation du véhicule *num* est composé de 9 caractères. L'heure d'arrivée est stockée en secondes (valeur entre 0 et 3600).

Attention ! Définissez le type de donnée *CharArray* comme un tableau de 9 caractères. Interdiction d'utiliser le type String !

Nous allons maintenant à écrire un ensemble minimal de sous-programmes qui implémentent les opérations de manipulation du *Type_Voiture*. N'oubliez de proposer (en parallèle avec l'implémentation des opérations) un programme principal *test.adb* qui assure le comportement attendu des sous-programmes.

3. Écrire le sous-programme *lire.voiture* qui lit les informations d'une voiture *v* de type *Type_Voiture*. L'heure en secondes *heureEntree* est calculée à partir de l'heure et les minutes de l'heure d'entrée (*heureEntree_h* et *heureEntree_m*). Le programme doit assurer les préconditions sur les valeurs *heureEntree_m* et *heureEntree_h*. Les figures ci-dessous montrent différents résultats attendus à l'écran après l'exécution du sous-programme *lire.voiture* :

```
Introduire les 9 caracteres correspondant au numero d'immatriculation: 23456734A
Introduire la place du parking : 34
Introduire l'heure d'arrivee : 0
Introduire les minutes de l'heure d'arrivee : 59
```

```
Introduire les 9 caracteres correspondant au numero d'immatriculation: 345678AAA
Introduire la place du parking : 5
Introduire l'heure d'arrivee : -2
La valeur saisie n'est pas valide
Introduire une nouvelle valeur pour l'heure d'arrivee : 3
Introduire les minutes de l'heure d'arrivee : 67
La valeur saisie n'est pas valide
Introduire une nouvelle valeur pour les minutes de l'heure d'arrivee : 3
```

4. Écrire le sous-programme *afficher.voiture* qui affiche toutes les informations d'une voiture *v* de type *Type_Voiture*. La figure ci-dessous montre le résultat attendu à l'écran après l'exécution du sous-programme *afficher.voiture* :

```
##### Informations #####
La voiture avec numero d'immatriculation : 234567EEE
est stationee a la place de parking : 3
et son heure d'arrivee est : 2h: 45m
```

5. Modifiez le programme *test.adb* afin de valider les sous-programmes *lire.voiture* et *afficher.voiture*.
6. Écrire le sous-programme *egalite* (ou opérateur "="). Ce sous-programme permet de comparer les informations de deux voitures *v1* et *v2* de type *Type_Voiture*. L'opération retourne VRAI si les deux voitures ont les mêmes numéros d'immatriculations sans prendre en compte la place du parking ni l'heure d'arrivée.

Attention! Le sous-programme *egalite* utilise une boucle pour comparer individuellement chaque caractère du numéro d'immatriculation.

Pour valider cette opération, vous devez effectuer la vérification sur plusieurs scénarios possibles (deux voitures avec même numero d'immatriculation, ...)

7. Écrire le sous-programme *obtenir.matricule* qui retourne la valeur de matricule d'une voiture *v* de type *Type_Voiture*. Modifiez le programme *test.adb* afin de

valider le sous- programme *obtenir_matricule*.

8. Écrire le sous-programme *obtenir_num_place* qui retourne la valeur de place de stationnement d'une voiture *v* de type *Type_Voiture*. Modifiez le programme *test.adb* afin de valider le sous- programme *obtenir_num_place*.
9. Écrire le sous-programme *calculer_duree_stationnement* qui calcule la durée de stationnement (en secondes) d'une voiture *v* de type *Type_Voiture*. Le calcul est fait en fonction de l'heure d'entrée *heureEntree* et de sortie d'un parking. L'heure et les minutes de l'horaire de sortie est une information en entrée du sous-programme (*heureSortie_h* et *heureSortie_m*). Le sous-programme renvoie -1 si l'heure de sortie est inférieur à l'heure d'arrivée. Modifiez le programme *test.adb* afin de valider le sous- programme *calculer_duree_stationnement*. Pour valider le sous-programme, vous devez effectuer la vérification dans différents scenarios.