

Tp NetCar

TRAVAIL À FAIRE	
1	Démarrer l'application <i>tpNetCar</i> . Saisir un jeu d'essai composé des trois types d'entretien et d'une voiture.

TRAVAIL À FAIRE	
2	Un test sur la fonctionnalité de « vérification des entretiens d'un véhicule » (CASE 7 de l'application) lève une exception lorsque l'immatriculation du véhicule saisie n'est pas référencée comme l'illustre la capture d'écran ci-dessous. En vous appuyant sur l'outil de débogage, implémentez une solution pour résoudre ce problème.

```
38 public bool NecessiteEntretien(Vehicule unVehicule)
39 {
40     bool necessaire = false;
41     int nbKmActuel = unVehicule.ObtenirNombreDeKilometreAuCompteurDuVehicule(); //kilométrage du véhicule
42     Entretien dernierEntretien = unVehicule.ObtenirEntretien(unVehicule.ObtenirNombreDesEntretiensRealises(unVehicule));
43     int nbKilometreDernierEntretien = dernierEntretien.NbKilometre;
44     int kmParcoursDepuisDernierEntretien = nbKmActuel - nbKilometreDernierEntretien;
45     foreach (EntretienType e in CollectionEntretien)
46     { //on parcourt les entretiens
47         int kmEntretien = e.ObtenirNombreDeKilometre;
48         int kmTolerance = e.ObtenirMargeDeTolerance();
49         //si le nombre de kilomètre parcouru se situe entre les marges de tolérance
50         if (kmParcoursDepuisDernierEntretien >= kmTolerance && kmParcoursDepuisDernierEntretien <= kmEntretien)
51         {
52             necessaire = true;
53             return true;
54         }
55     }
56     return false;
57 }
```

System.NullReferenceException : 'La référence d'objet n'est pas définie à une instance d'un objet.'

unVehicule a été null.

Variables locales

Nom	Valeur	Type
\$exception	("La référence d'objet n'est pas définie à une instance d'un objet.")	System.NullReferenceException
this	(tpCasNetCar.Atelier)	tpCasNetCar.Atelier
unVehicule	null	tpCasNetCar.Vehicule
necessaire	false	bool
nbKmActuel	0	int
unVehicule	null	tpCasNetCar.Vehicule
nbKilometreDernierEntretien	0	int
kmParcoursDepuisDernierEntretien	0	int

```
Vehicule v = unAtelier.RechercheVehicule(immatriculation)
if(v != null)
{
    bool necessaire = unAtelier.NecessiteEntretien();
}
```

TRAVAIL À FAIRE	
3	Un test sur la fonctionnalité d'« ajout d'un véhicule » (CASE 1 de l'application) a révélé qu'il était possible de saisir plusieurs fois le même véhicule, c'est-à-dire le même numéro d'immatriculation. Implémentez une solution qui rend impossible un tel cas de figure.

TRAVAIL À FAIRE	
4	Complétez la partie case 5 de l'application relatif à la suppression d'un entretien type à l'atelier.

TRAVAIL À FAIRE	
5	<p>La classe Program comporte une méthode à portée classe, nommée <i>IsImmatriculation</i>, qui vérifie le format d'une immatriculation.</p> <p>Ecrire une méthode à portée classe qui permette la saisie d'une immatriculation à l'instar de la méthode <i>SaisirEntier</i>. Cette méthode s'appuiera sur la méthode <i>IsImmatriculation</i> pour effectuer un contrôle de saisie.</p> <p>Par ailleurs, les tests effectués à l'aide de la méthode <i>IsImmatriculation</i> ne sont pas concluants. Exemples de tests d'immatriculation effectués :</p> <p>AB-985-QM // false devrait être true AB985QM //true devrait être false A-985-XR //false Ok</p> <p>Modifiez le pattern de l'expression régulière afin de respecter le format attendu : AA-999-AA</p>