

**OBJECTIFS**

- Savoir définir et utiliser une méthode
- Distinguer responsabilités entre la méthode et le programme appelant

**A retenir !**

- Les méthodes dédiées à l’affichage et ou à la saisie (5 %) peuvent contenir des appels à WriteLine, ReadLine.
- Toutes les autres **méthodes (95%)** doivent contenir uniquement des traitements et calculs : elles **ne doivent pas contenir** des appels à **WriteLine, ReadLine**.

**EXO 1 :**

1. Au sein du répertoire « **Sequence\_3\_Methodes** », créez un projet « **Exo\_CourseAPied** » dans une solution :  
« **Seance3\_Definition\_Methodes\_Approfondissement** »
2. En dessous du main , définissez la méthode :  
**public static double CalculeVitesse (double nbKm, int nbMinutes) :**  
retourne le nombre de km/h arrondi à un chiffre après la virgule.

Données en entrée		Résultat
nbKm	nbMinutes	kmParHeure
6,5	45	8,7

**Pensez à tester la validité des paramètres avant tout traitement. Ex :**

```
if (... )
    throw new ArgumentOutOfRangeException ("Attention, le nb de km doit être");
```

3. Au sein du main, demandez à l'utilisateur le nb de km parcourus, le temps qu'il y a passé. Utilisez la méthode précédemment définie et affichez sa vitesse.

```
-----
COURSE A PIED
-----
Nb de km parcourus :
6,5
Temps (en min):
45
-----
VITESSE :
-----
8,7 km/h
```

4. Améliorez votre programme : Votre programme fonctionne pour les coureurs de moins d'une heure ! si le coureur court 1h ou plus, il ne doit pas être obligé de faire la conversion. Définissez alors la méthode

**public static int ConvertitTempsEnMinutes( String temps)**

retourne le nombre de minutes correspondant à la chaîne temps spécifiée .

Aide : utilisez Substring et Parse

**Pensez à tester la validité des paramètres avant tout traitement.**

Données en entrée	Résultat
temps	nbMin
01 :25	85

5. Au sein du main, modifiez :

- le format attendu pour la saisie du temps (en min) => (hh:mm)
- la saisie

Puis utilisez votre méthode pour convertir la saisie faite par l'utilisateur. Affichez l'équivalence ainsi que la vitesse . Ex :

```
-----
COURSE A PIED
-----
Nb de km parcourus :
12
Temps (hh:mm):
01:25
<=> 85 min
-----
VITESSE :
-----
8,5 km/h
```

6. Finalement, vous allez placer ces 2 méthodes dans une nouvelle classe « **Course** ». Faites les modifications nécessaires dans le main.
7. Ajoutez à la classe Course une nouvelle méthode :
- public static double NbKmPourAtteindreVitesse(int nbMin, double vitesse)**: retourne un arrondi du nb de km à faire pour atteindre la vitesse spécifiée en km/h durant les nbMin spécifiées.

Données en entrée		Résultat
nbMin	vitesse	nbKm
85	10	14,2

Au sein du main, demandez au coureur la vitesse qu'il désire atteindre puis utiliser la méthode pour indiquer les progrès à faire

```
-----
OBJECTIFS A ATTEINDRE :
-----
Vitesse à atteindre :
10
-----
Pour avoir une allure moyenne de 10km/h
- Avec le même temps, il faut faire 14,2km
progres de 2,2 km (en +).
```

8. Ajoutez à la classe Course une nouvelle méthode :

**public static int NbMinPourAtteindreVitesse(double nbKm, double vitesse)**

retourne le nb de minute à tenir pour atteindre la vitesse spécifiée en km/h sur le nb de km spécifié.

Données en entrée		Résultat
nbKm	vitesse	nbMin
12	10	72

Au sein du main, avec la vitesse à atteindre déjà saisie, indiquer les progrès à faire

```
- Avec la même distance, il faut faire 72min
  progres de 13 min (en-).
-----
```

9. Améliorez votre programme. Au lieu d'avoir le temps à atteindre en minutes, on désire le voir au format hh:mm. Pour cela, ajoutez à la classe Course une dernière méthode :

**public static String ConvertitMinutesEnTemps(int nbMinutes)**

Retourne une chaîne au format hh:mm à partir du nombre de minutes spécifiées.

Aide : pour transformer 1 en « 01 », il faut utiliser ToString, et lui préciser qu'on veut un format décimal de 2 chiffres : cela permet d'ajouter des 0 non significatifs.

```
int nb=1;
String txt = nb.ToString("D2") ;
```

#### POUR LES PLUS RAPIDES

Utilisez le type TimeSpan et ses méthodes déjà définies : Parse, FromMinutes et la propriété TotalMinutes : ne supprimez pas vos méthodes existantes, surchargez les si nécessaires. Reprenez votre code dans un projet « un projet » **Exo\_CourseAPied\_TimeSpan** »