

Programmation 1 bis

TP : Course de voitures

1 Introduction

On souhaite simuler une course de voitures. Pour cela, on crée les classes `Voiture`, `Course` et `MainCourse`, qui, comme leur nom l'indique, permettent de gérer les voitures, les courses de voitures et leur exécution respectivement.

La classe `Voiture`

Une voiture est caractérisée par son nom (chaîne de caractères), sa position sur un axe à partir d'un point origine, qui est un entier positif ou nul, son sens de déplacement sur cet axe (1 si elle s'éloigne de l'origine et -1 si elle s'en rapproche) et sa vitesse, qui est un entier positif ou nul. Pour la simulation à l'écran, l'axe sera horizontal et le point origine tout à fait à gauche de l'écran.

La classe `Voiture` a donc 4 attributs : `nom`, `position`, `sens` et `vitesse`. Elle possède le constructeur et les méthodes suivants :

```
public Voiture(String unNom, int uneVitesse)
/* pré-requis : (à compléter)
 * action : crée une voiture de nom unNom et de vitesse uneVitesse
 *          placée à l'origine, prête à partir
 */
public String toString()
/* résultat :  retourne une chaîne de caractères contenant les caractéristiques
 *              de this (sous la forme de votre choix)
 */
public String toString2()
/* résultat :  retourne une chaîne de caractères formée d'une suite d'espaces
 *              suivie de la première lettre du nom de this, suivie d'un retour
 *              à la ligne, le nombre d'espaces étant égal à la position de this.
 */
public String leNom()
/* résultat :  retourne le nom de this */

public boolean depasse(int limite)
/* résultat :  retourne vrai si et seulement si la position de this est
 *              supérieure ou égale à limite
 */
```

```

public void faitDemiTour()
/* action : fait faire un demi-tour à this */

public void avance()
/* pré-requis : (à compléter)
 * action : fait avancer this d'une distance égale à sa vitesse dans son sens de
 *          déplacement (à compléter)
 */

```

Vous devez préciser dans les spécifications de la méthode **avance** comment traiter le cas où **this** se dirige vers l'origine avec une vitesse strictement supérieure à sa position : cas à exclure dans les pré-requis ? Blocage de **this** à l'origine ? Rebondissement de **this** avec demi-tour à l'origine ?

La classe Course

Une course (entre 2 voitures) est caractérisée par les 2 voitures en question, qui sont des instances de la classe **Voiture**, ainsi que par la longueur de la piste, qui est un entier strictement positif. Le déroulement d'une course se passe de la façon suivante. Au début, les 2 voitures sont sur la ligne de départ. A chaque étape, l'une des 2 voitures, choisie aléatoirement, avance. La course s'arrête dès qu'une voiture a franchi la ligne d'arrivée.

La classe **Course** a donc 3 attributs : **voit1**, **voit2** et **longueurPiste**. Elle possède le constructeur et les méthodes suivants :

```

public Course (Voiture uneVoit1, Voiture uneVoit2, int longueur)
/* pré-requis : (à compléter)
 * action :      (à compléter)
 */
public String toString()
/* résultat :  retourne une chaîne de caractères contenant les caractéristiques
 *             de this (sous la forme de votre choix)
 */
public Voiture deroulement()
/* pré-requis : this.voit1 et this.voit2 sont sur la ligne de départ
 *             (à l'origine), prêtes à partir
 * action :    simule le déroulement d'une course entre this.voit1 et this.voit2
 *             sur une piste de longueur this.longueurPiste, en affichant à chaque
 *             étape les 2 voitures, représentées par la première lettre de leur
 *             nom, à leur position respective, et retourne la voiture gagnante.
 */

```

La classe MainCourse

La classe **MainCourse** permet d'exécuter une course entre 2 voitures. Elle crée deux instances **v1** et **v2** de la classe **Voiture**, puis une instance de la classe **Course** permettant à **v1** et **v2** de faire une course, puis affiche le déroulement d'une course entre **v1** et **v2** et affiche finalement le nom de la voiture gagnante.

Conseil : dans la console, pour simuler l'évolution des voitures, vous pouvez utiliser les procédures `Ut.clearConsole()` et `Ut.pause(int timeMilli)` entre deux étapes de la course.

Extensions possibles

Pour une course

1. Donner aux 2 voitures des vitesses différentes et des probabilités différentes d'être choisies pour avancer (en équilibrant les chances pour une course équitable!).
2. Défavoriser une voiture au départ, mais pour équilibrer les chances, permettre à une voiture d'accélérer quand son écart avec la voiture de tête devient trop important (pour cela, il vous faudra rajouter des méthodes à la classe **Voiture** pour permettre à une voiture d'accélérer et pour pouvoir évaluer l'écart entre 2 voitures).
3. Choisir comme itinéraire de la course un (ou plusieurs) aller-retour(s) entre les 2 extrémités de la piste.
4. Simuler une course entre un nombre quelconque de voitures (utiliser un tableau d'instances de la classe **Voiture**).

Pour une voiture

1. Améliorer l'affichage à l'écran en utilisant les possibilités graphiques de Java (voir le répertoire `prog1Bis/Utile/graphisme.zip` sur l'ENT).
2. Faire évoluer la voiture dans un plan : remplacer l'attribut `position` par les coordonnées `x` et `y` et l'attribut `sens` par une direction pouvant avoir, soit 4 valeurs possibles (les 4 points cardinaux), soit 360 valeurs possibles si vous souhaitez permettre toutes les directions possibles.