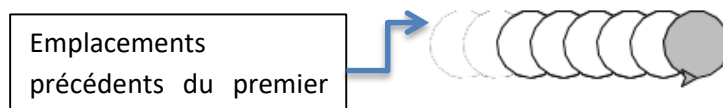


Les chenilles animées¹

1. Objectif

On souhaite simuler le déplacement d'une chenille sous une forme proche de la suivante :



Une chenille est constituée de différents anneaux avec un rayon défini.

On suppose que la chenille se dirige toujours de façon parallèle à Ox orientée vers la droite et d'une distance d'un *rayon* à chaque "pas".

A l'instant initial, la chenille démarre en $(0, 0)$.

On vous demande de programmer le déplacement de chenilles en mettant en œuvre la programmation orientée objet en suivant les étapes ci-dessous.

2. Les anneaux

Un anneau est un *Cercle* particulier. Pour ce faire, on peut utiliser la définition de la classe *Cercle* précédemment définie dans l'exercice sur les formes géométriques (attributs : rayon, couleur, position).

Un anneau a un attribut de plus qu'un cercle, la couleur utilisée pour dessiner le bord de l'anneau, qui est obligatoirement noire.

Définir la classe *Anneau* et la méthode *dessiner* qui lui correspond.

Définit la méthode *effacer* qui efface l'anneau en dessinant un cercle blanc de même rayon et de même centre.

3. La tête

La tête de la chenille est un anneau particulier de couleur grise et de circonférence noire.

Définir la classe *Tete* et la méthode *dessiner* qui lui correspond, sachant que cette méthode dessine un œil sur la tête :



4. La chenille

Définir la classe *Chenille*, sachant que chaque instance de la classe *Chenille* a les attributs suivants :

- une position de départ (position de l'anneau le plus à gauche)
- une tête définie par son diamètre
- plusieurs anneaux qui peuvent avoir des couleurs de remplissage différentes et qui ont le même diamètre que la tête

4.1. Dessin d'une chenille

Ecrire la méthodes *dessiner* de cette classe.

4.2. Déplacement d'une chenille

On souhaite simuler le déplacement d'une chenille.

On suppose que la chenille se dirige toujours de façon parallèle à Ox orientée vers la droite et d'une

¹ Exercice inspiré de : <http://lig-membres.imag.fr/genoud/ENSJAVA/tds/sujets/tdChenille/tpChenille.html>

distance d'un rayon à chaque "pas".

a) Définir une méthode avance qui déplace la chenille en modifiant sa position d'une valeur de dx fournie en paramètre.

b) Ecrire un programme qui fasse avancer la chenille de nbp "pas", nbp fourni en paramètre.

c) Modifier le programme pour qu'il fasse avancer la chenille jusqu'à ce que la tête rencontre le bord droit de la fenêtre. Indications : fonction de turtle utile `window_width()`

5. Variantes

Il est possible d'imaginer plusieurs variantes :

- Création d'une sous-classe `ChenilleProcessionnaire` qui avance en « ondulant » (selon une sinusoïde)
- Création d'une classe `Course`, qui simule une course de plusieurs chenilles positionnées à différentes valeurs de y et ayant des vitesses tirées au sort (la vitesse étant le nombre de déplacements de la chenille par « pas de temps » de la course). La chenille gagnante étant la première arrivée au bord droit de la fenêtre.
- Etc.