

Semaine Tortue

Matière

- Environnement de travail (IntelliJ)
- Structure générale d'un programme en Java
- La séquence
- La répétitive *for* (1ère approche)
- L'appel de méthodes (1ère approche)

Objectif

- Savoir écrire un programme qui fait un dessin avec appel de méthodes données
- Savoir éditer un tel programme, le compiler, l'exécuter et le mettre au point

Si vous n'avez pas fait les puzzles proposés ce lundi, faites la série :

<https://studio.code.org/s/20-hour/stage/5/puzzle/1>

Exercices obligatoires

**Pour les exercices de cette semaine, créez le projet IntelliJ
ALGO_tortue**

La classe *Tortue* va vous permettre d'être un artiste en Java !

Elle possède les méthodes :

```
avancer(double pixels)
```

```
accélérer()
```

```
//Si la tortue est vraiment trop lente :  
//tortue.accélérer();
```

```
tournerADroite(int degres)
```

```
tournerAGauche(int degres)
```

```
definirCouleur(String couleur)
```

```
//Les couleurs connues sont : BLEU, VERT, JAUNE, NOIR, BLANC et ROUGE  
//La couleur "ALÉATOIRE" définit une couleur au hasard !
```

```
//Exemple :
```

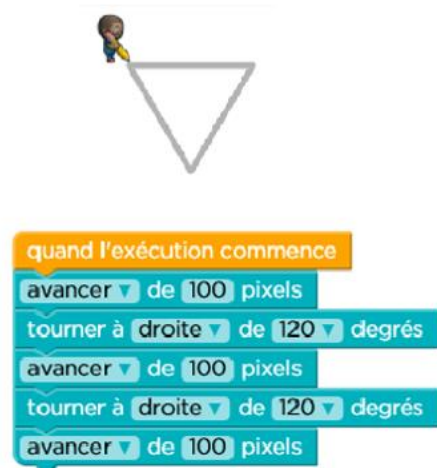
```
//tortue.definirCouleur("VERT");
```

A Dessins **sans** boucle for

Une fois l'exécution lancée, un programme exécute ses instructions l'une après l'autre en commençant par le haut. C'est ce qu'on appelle la « séquence ».

Voici comme exemple le dessin d'un triangle et la classe Java correspondante :

```
public class DessinTriangle {  
  
    public static void main(String[] args){  
  
        Tortue tortue = new Tortue();  
  
        tortue.avancer( pixel: 100);  
        tortue.tournerADroite( degre: 120);  
        tortue.avancer( pixel: 100);  
        tortue.tournerADroite( degre: 120);  
        tortue.avancer( pixel: 100);  
  
    }  
}
```



A1 Ecrivez la classe *DessinU* qui dessine la lettre U.

A2 Ecrivez la classe *DessinV* qui dessine la lettre V.

A3 Ecrivez la classe *DessinX* qui dessine la lettre X.

Si vous avez du mal à concevoir ce dessin, refaites le puzzle 8 de la série Artiste :
<https://studio.code.org/s/20-hour/lessons/5/levels/8>

Pour reculer, mettez un nombre négatif comme paramètre de la méthode `avancer()` !

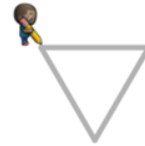
On peut repasser sur des traits.

B Dessins **avec** boucle for

Il arrive très souvent que l'on veuille répéter une série d'instructions un certain nombre de fois.

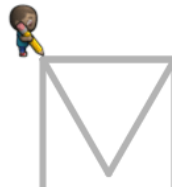
Voici à nouveau, comme exemple, le dessin d'un triangle et la classe Java correspondante :

```
public class DessinTriangle {  
  
    public static void main(String[] args){  
  
        Tortue tortue = new Tortue();  
  
        for (int i = 0; i < 3; i++) {  
            tortue.avancer( pixel: 100);  
            tortue.tournerADroite( degre: 120);  
        }  
  
    }  
  
}
```



B1 Ecrivez la classe *DessinCarre*.

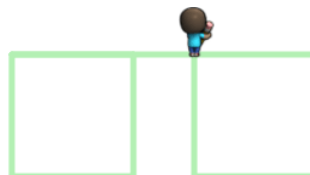
B2 Ecrivez la classe *DessinEnveloppe*.



B3 Ecrivez la classe *DessinMaison*.



B4 Ecrivez la classe *DessinLunettes*.



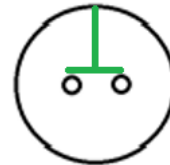
B5 Ecrivez la classe *DessinPriseElectrique*.

Il y a moyen de dessiner des très petits cercles.
Essayez :

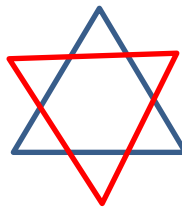
```
for (int i = 0; i < 360; i++) {  
    tortue.avancer(0.1);  
    tortue.tournerADroite(1);  
}
```



Il est possible de déplacer la tortue en choisissant la couleur
noire. Le fond de l'écran est noir.
Du noir sur du noir, cela ne se voit pas !



B6 Ecrivez la classe *DessinEtoile*.



Exercices supplémentaires

C Dessins **avec** boucle for **dans** une boucle for

C1 Ecrivez la classe *DessinFleur*.

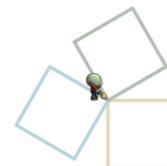
Commencez par faire le puzzle :

<https://studio.code.org/s/20-hour/stage/7/puzzle/3>

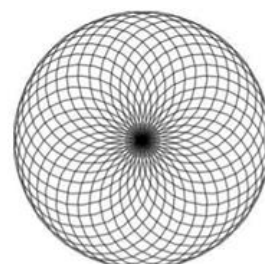


C2 Ecrivez la classe *DessinVentilateur*.

Réfléchissez ! Il y a 3 carrés et un carré possède 4 côtés...



C3 Ecrivez la classe *DessinSpirographe* qui dessine :



Exercices défis

Voici quelques idées de dessin défi :

