

1 Point2D

Création d'une classe

Ecrire une nouvelle classe **Point2D**

Elle devra respecter les contraintes suivantes :

- 2 attributs x et y de type *int*
- Une méthode pour modifier la valeur de x (**void setX(int valX)**)
- Une méthode pour modifier la valeur de y (**void setY(int valY)**)
- Une méthode pour récupérer la valeur de x (**int getX()**)
- Une méthode pour récupérer la valeur de y (**int getY()**)
- Une méthode pour afficher le point (**void afficher()**). Cette méthode affichera les valeurs des attributs x et y sur la sortie standard. Par exemple sous cette forme :

```
[4, 34]
```

- Une méthode pour traduire le point (**void traduire(int dX, int dY)**). On est ici sur une [translation vectorielle](#), la valeur de x devient x+dX et la valeur de y se transforme en y+dY.

Point2D
-x: int -y: int
+afficher(): void +getX(): int +getY(): int +setX(valX: int): void +setY(valY: int): void +traduire(dX: int, dY: int): void

Test

Ecrire une nouvelle classe **Run** dans le dossier « Jour01 » qui devra respecter les contraintes suivantes :

- avoir une méthode **public static void main(String[] args)**
- Dans la méthode main, créez deux variables locales serviront de x et de y

Ajout de constructeurs

Reprenez votre classe Point2D.

Ajoutez deux constructeurs qui respectent les contraintes suivantes :

- le premier ne prendra aucun paramètre, initialisera x et y à zéro

- le second prendra en paramètre une valeur d'initialisation pour x et une valeur d'initialisation pour y (**Point2D(int vX, int vY)**)

Chaîner les constructeurs

Point2D
-x: int -y: int
«constructor»+Point2D() «constructor»+Point2D(vX: int, vY: int) +afficher(): void +getX(): int +getY(): int +setX(valX: int): void +setY(valY: int): void +translater(dX: int, dY: int): void

Test

Reprenez votre classe Run et faites usage de vos nouveaux constructeurs

Ajout d'un attribut static

Reprenez votre classe Point2D.

Ajoutez un attribut à cette classe qui devra compter le nombre d'objet créé dans cette famille (*la famille des Point2D*).

L'attribut étant commun à toutes les instances de la classe Point2D il sera **static**, son type sera un entier.

Incrémentez sa valeur dans les constructeurs de la classe Point2D.

Ajoutez une méthode qui devra rendre la valeur de l'attribut, ce dernier étant static, la méthode sera aussi static.

Test

Reprenez votre classe Run et vérifiez que votre nouvel attribut compteur fonctionne.

2 Time

Création d'une structure

Créez une nouvelle structure nommée Time

Elle doit permettre de stocker une heure de la journée comme 10:05 ou 00:45 sous a forme du nombre de minutes depuis minuit.

Insérez une méthode `afficher()`, qui permet d'afficher la valeur de la structure (hh:mm)

Exemple : Si je crée une instance de la class `Time` avec 45 minutes, la fonction `afficher()` doit m'afficher « 00:45 », je crée avec 605 minutes, la fonction `afficher()` doit m'afficher « 10:05 » etc...

Ecrire une nouvelle classe **Run** qui permettra de créer des variables de type `Time` et afficher la valeur de ces variables