Exo 1:

1. Il écrit la commande pour imprimer (print) :

System.***out***.println();

1. Il nous donne plusieurs options et quand on choisi la toString(), il écrit :

@Override

**public** String toString() {

// **TODO** Auto-generated method stub

**return** **super**.toString();

}

1. Il écrit la méthode pour main :

**public** **static** **void** main(String[] args) {

}

1. Il nous donne plusieurs options dont : constructor, equals function, toString() function, …

**public** First() {

// **TODO** Auto-generated constructor stub

}

@Override

**public** **boolean** equals(Object obj) {

// **TODO** Auto-generated method stub

**return** **super**.equals(obj);

}

@Override

**public** String toString() {

// **TODO** Auto-generated method stub

**return** **super**.toString();

}

1. 6. Nous pouvons renommer la classe et toutes les références qui font référence à cette classe dans le fichier. Idem pour le champ foo.

Exo 2:

1. Il fonctionne avec default, on a un constructeur por point qui prend pas de variables

2. erreur de compilation, il marche pas, parce qu’on a une variables private x.

pour cela on peut les rendre public x et y, ou en utilisant un getter

3. Pour protéger les valeurs ou la fonction des actions indésirables, ce qui signifie que l'appelant ne peut généralement pas obtenir un accès direct inapproprié au champ  
4. Les valeurs de ce champ dans une autre classe. Nous utilisons donc une méthode pour obtenir la valeur du champ. Accessor est une getter qui est généralement utilisée lorsque nous avons un champ privé et que nous voulons  
Oui, nous devons le faire ici car nous avons des valeurs privées (x et y) et nous devons y accéder à partir d'une autre fonction dans une autre classe  
5. Nous avons une erreur de compilation car nous avons maintenant un constructeur prenant des variables et le constructeur par défaut que nous avions auparavant n'est pas celui par défaut3. Pour protéger les valeurs ou la fonction des actions indésirables, ce qui signifie que l'appelant ne peut généralement pas obtenir un accès direct inapproprié au champ  
4. Les valeurs de ce champ dans une autre classe. Nous utilisons donc une méthode pour obtenir la valeur du champ. Accessor est une getter qui est généralement utilisée lorsque nous avons un champ privé et que nous voulons  
Oui, nous devons le faire ici car nous avons des valeurs privées (x et y) et nous devons y accéder à partir d'une autre fonction dans une autre classe  
5. Nous avons une erreur de compilation car nous avons maintenant un constructeur prenant des variables et le constructeur par défaut que nous avions auparavant n'est pas celui par défaut.

6. Le programme sera confus entre les valeurs que nous avons dans la classe et les valeurs que nous passons en paramètres, nous devons donc utiliser le mot-clé this pour définir les valeurs que nous avons dans la classe  
7. Créez une variable appelée somme dans la classe et incrémentez sa valeur dans les constructeurs de point  
8. Si on passe par la fonction deux doubles en paramètres, le compilateur appellera cette méthode :

**public** Point(**double** x,**double** y) {

et si on passe un seul point en paramètre, le constructeur appellera cette méthode :

**public** Point(Point p2) {

Exo 3:

1. Output:

True

False

Parce que dans le premier pointent tous les deux vers le même endroit en mémoire mais dans le second chaque objet pointe vers un endroit différent en mémoire même s'ils ont tous les deux les mêmes valeurs pour x et y

1. Output:

0

-1

Exo 4:

2. Si nous ajoutons plus de points de la capacité maximale, nous aurons une erreur de temps d'exécution  
Nous devons lancer une exception pour que, lorsque nous atteignons la capacité maximale, nous ne puissions plus ajouter de valeurs  
5. Si nous ajoutons null, cela ajoutera null dans le tableau.  
6. Dans un arrayList, nous n'avons pas de limite à la taille de la liste. Ainsi, pointCapacity est illimité, nbPoints sera égal à la taille de la liste et la méthode contient ne changera pas car nous utilisons le for chaque boucle.

Exo 5:

1. on peut incrémenter les valeurs de x et y en ajoutant la valeur de dx à x et la valeur de dy à y  
5. nous utilisons le point p pour les deux cercles, donc lorsque nous traduisons un cercle, nous modifions la valeur de p, donc les deux cercles se traduisent également. Pour l'éviter, nous devons créer un nouveau point j avec les nouvelles valeurs de x et y, puis assimiler p à j.  
9. Cette méthode doit être déclarée dans la classe TestPoint et non dans la classe Circle.

Exo 6:

1. Non

4. Il va afficher l'addresse in memory of ring and not the actual data that we want, we need to add a toString method