

Programmation orientée objet

TP4

Romain Marie & Hervé Gaudin

Octobre 2019

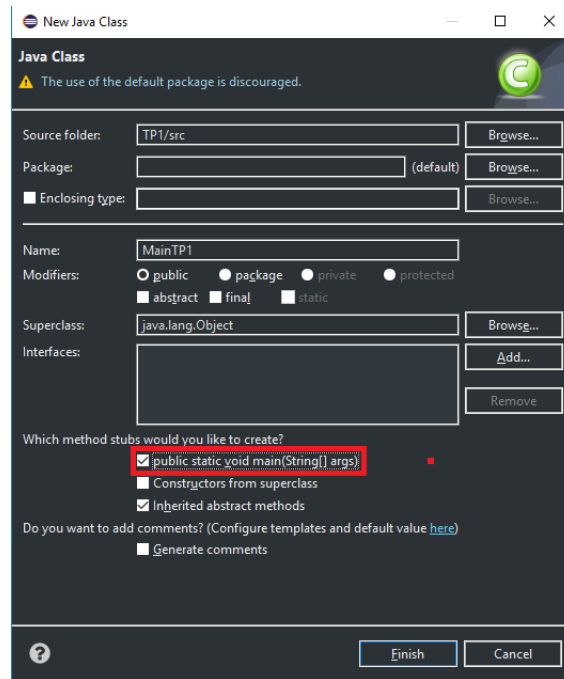


1 Programme, point d'entrée, Hello world

Pour rappel, un projet Java se découpe en un ensemble de fichiers qui vont contenir les classes de votre programme.

Pour qu'un programme soit exécutable, il doit respecter les conditions suivantes :

- Posséder au moins une classe (pour rappel, en java, tout est codé à l'intérieur de classes). Par convention, le nom d'une classe commencera toujours par une majuscule
 - Posséder un point d'entrée, c'est à dire une méthode nommée *main()* décrivant la suite d'instructions à réaliser par votre programme. En clair, votre programme commencera à la première ligne de votre point d'entrée, et s'achèvera lorsque sa dernière ligne sera exécutée.
1. Créez un projet Java que vous nommerez "TP4". Vous prendrez garde à ne pas créer le fichier "module-info.java". (ou à le supprimer si vous vous êtes trompés).
 2. Ajoutez une classe à votre projet que vous nommerez **MainTP4**. Au moment de sa création, et tel qu'illustré ci-dessous, cochez la case vous permettant de créer automatiquement la méthode *main()*.



3. Remplissez la méthode `main()` pour que votre programme affiche à l'écran "Bienvenue dans le TP 4".
4. Exécutez votre programme pour vérifier que l'affichage s'effectue correctement

2 Un peu de classe

1. Ajoutez une classe `Cercle` à votre projet ! Cette fois, elle ne devra pas posséder de méthode `main()`, puisque le point d'entrée de votre programme est défini dans la classe `MainTP4`.
2. Ajoutez les propriétés suivantes à votre classe (elles devront être définies comme privées) :
 - un rayon r de type `double`
 - une coordonnée x et une coordonnée y de type `double`
3. Ajoutez les méthodes suivantes à votre classe :
 - Une méthode `void initialiser(int X, int Y, int R)` permettant de spécifier les propriétés du `Cercle`. Bien sûr, il faudra que le `Cercle` ait un rayon positif.
 - Une méthode `void deplacer(int X, int Y)` qui centrera le `Cercle` aux coordonnées passées en paramètre.
 - Une méthode `double calculAire()` qui calcule et retourne l'aire du cercle. Vous noterez que cette méthode ne prend aucun paramètre.
 - Une méthode `void grossir(double delta)` qui augmente le rayon du `Cercle` de la valeur δ passée en paramètre.
 - Une méthode `String toString()` qui renvoie une chaîne de caractère décrivant les propriétés du `Cercle` sous la forme `"(r, x, y)"` (où r , x et y sont remplacés par leurs valeurs).
4. Dans votre point d'entrée (méthode `main()` de la classe `MainTP1`) :
 - (a) Créez un `Cercle c` de rayon 1 et de centre (0,0).
 - (b) Créez un second `Cercle c2` de rayon 3 centré en (2,1).
 - (c) Affichez l'aire du cercle c .

- (d) Affichez les propriétés du cercle *c2*.
 - (e) Faites grossir le cercle *c2* d'une valeur de 3, puis réaffichez ses propriétés.
5. Ajoutez à votre classe Cercle une méthode *collision(???)* capable d'indiquer si le cercle est en collision avec un autre cercle passé en paramètre.
 6. Affichez si *c* et *c2* sont en collision.
 7. Créez un tableau de 10 Cercle. Remplissez le de telle sorte que chaque Cercle ait un rayon aléatoire entre 1 et 5, et un centre de coordonnées aléatoires entre 1 et 10.
 8. Affichez les couples de Cercle de ce tableau qui sont en collision.