

Classes imbriquées dans Java

Lorsqu'on décrit une classe dans une autre classe dans Java sans le faire dans une méthode (car oui on peut le faire dans une méthode, mais ce ne sera pas abordé ici), vous pouvez faire une class avec ou sans le mot clé static.

- **Dans les 2 cas, le code de la classe est dans la zone de code.**
- **La classe englobante peut toujours avoir accès aux membres (static et non static) des classes imbriquées peu importe si la classe imbriquée est private/public /protected /rien**

Classes imbriquées static (static class ClassName{...})

Exemple :

```
public class Outer{  
    private static class Inner{  
        ....  
    }  
}
```

- N'a pas besoin d'une instance de la classe imbriquante pour être instanciée.
- Ne contient pas de pointeur vers la classe imbriquante
- N'accède pas aux membres d'instance de la classe imbriquante (méthodes et variables) peu importe l'accès
- Peut accéder aux membres static de la classe imbriquante (méthodes et variables) peu importe l'accès

Classes imbriquées non static (class ClassName{...})

Exemple :

```
public class Outer{  
    private class Inner{  
        ....  
    }  
}
```

- A besoin d'une instance de la classe imbriquante pour être instanciée
- Contient un pointeur vers la classe imbriquante
- Accède à tout de la classe imbriquante (peu importe l'accès) :
 - aux membres d'instance
 - aux membres static
 - aux méthodes d'instance
 - aux méthodes static
 - même aux private