

# EXPOSICIÓN DE POO

KEVIN ALAN  
MAYTE  
ANGEL DE JESUS

## CLASE ANIDADAS O NESTED CLASSES

Las clases anidadas te permiten agrupar lógicamente clases que solo se utilizan en un lugar, por lo tanto, esto aumenta el uso de la encapsulación y crea un código más fácil de leer y de mantener.



# CARACTERISTICAS

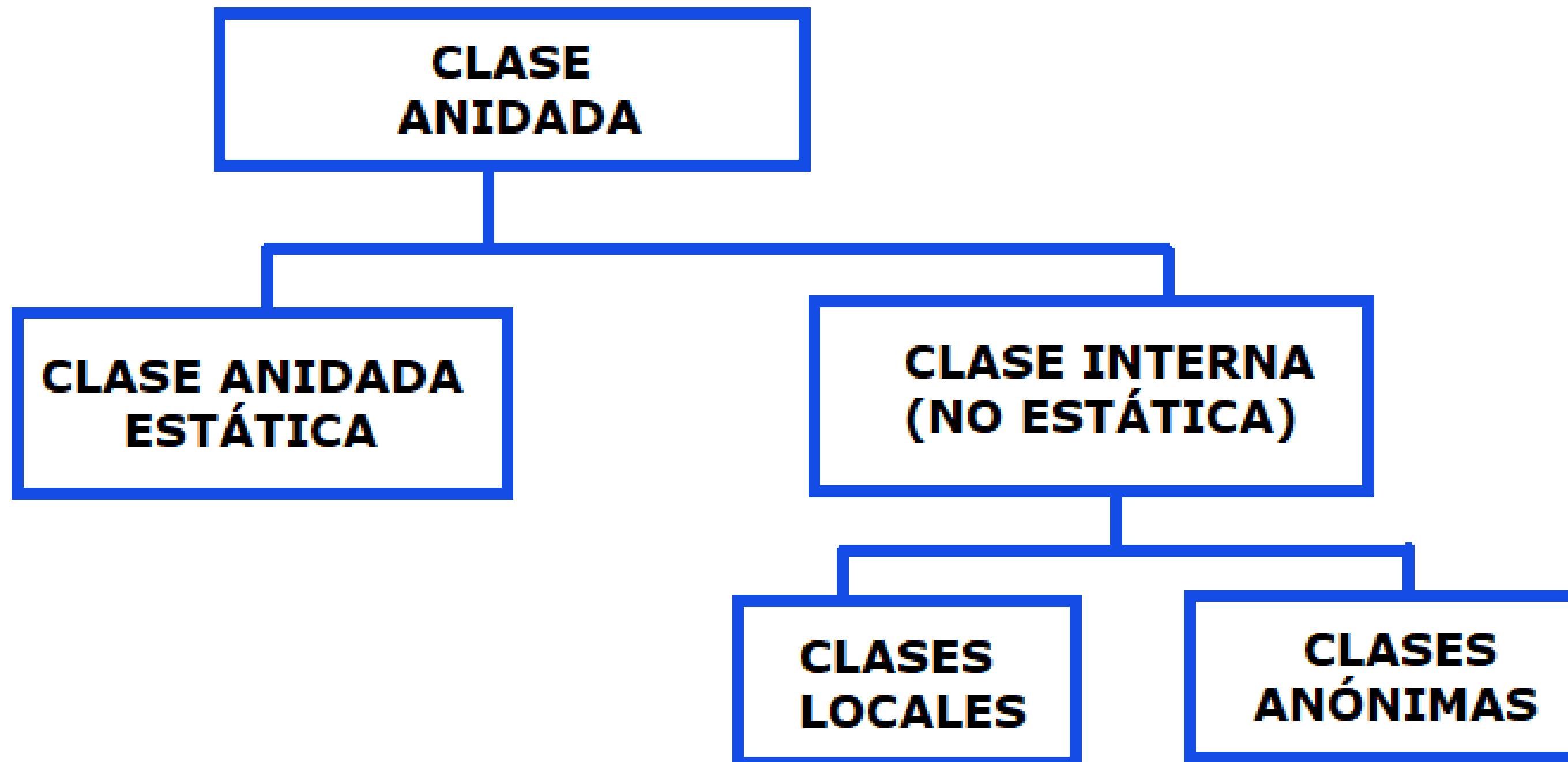


- Una clase anidada no existe independientemente de su clase adjunta. Por lo tanto, el alcance de una clase anidada está limitado por su clase externa.
- Una clase anidada también es miembro de su clase adjunta. También es posible declarar una clase anidada que es local a un bloque.
- Una clase anidada tiene acceso a los miembros, incluidos los miembros privados, de la clase en la que está anidado. Sin embargo, lo inverso no es verdadero, es decir, la clase adjunta no tiene acceso a los miembros de la clase anidada.





HAY DOS TIPOS GENERALES DE CLASES ANIDADAS: LAS QUE ESTÁN PRECEDIDAS POR EL MODIFICADOR STATIC (STATIC NESTED CLASS) Y LAS QUE NO LO ESTÁN (INNER CLASS).



# CLASES ANIDADAS

## Estáticas

Al igual que con los métodos y variables de clase, una clase anidada estática está asociada a su clase externa. Y al igual que los métodos de clase estáticos, una clase anidada estática no puede hacer referencia directamente a variables de instancia o métodos definidos en su clase adjunta: solo puede usarlos a través de una referencia de objeto.

```
ClaseExterna.ClaseAnidadaStatic objetoAnidado =  
new ClaseExterna.ClaseAnidadaStatic();
```

## Internas

Una clase interna tiene acceso a todas las variables y métodos de su clase externa y puede referirse a ellos directamente de la misma manera que lo hacen otros miembros no estáticos de la clase externa.

Para crear una instancia de una clase interna, primero debe crear una instancia de la clase externa. Luego, crea el objeto interno dentro del objeto externo con esta sintaxis:

```
ClaseExterna objetoExterno=new ClaseExterna();  
ClaseExterna.Claselnterna objetoInterno=  
objetoExterno.new Claselnterna();
```

# DIFERENCIA ENTRE CLASES ESTÁTICAS E INTERNAS (ANIDADAS NO ESTÁTICAS)



Las clases anidadas estáticas no tienen acceso directo a otros miembros (variables y métodos no estáticos) de la clase adjunta porque, como es estática, debe acceder a los miembros no estáticos de su clase adjunta a través de un objeto. Debido a esta restricción, las clases anidadas estáticas rara vez se utilizan.



Las clases anidadas no estáticas, tienen acceso a todos los miembros (variables y métodos estáticos y no estáticos, incluido el privado/private) de su clase externa y pueden referirse a ellos directamente de la misma manera que otros miembros no estáticos del exterior de la clase.

# CLASES INTERNAS ANÓNIMAS

LAS CLASES ANÓNIMAS SON MUY SIMILARES A LAS CLASES INTERNAS LOCALES, PERO SIN NOMBRE. EN LAS CLASES INTERNAS LOCALES PRIMERO SE DEFINE LA CLASE Y LUEGO SE CREAN UNO O MÁS OBJETOS. EN LAS CLASES ANÓNIMAS SE UNEN ESTOS DOS PASOS: COMO LA CLASE NO TIENE NOMBRE SÓLO SE PUEDE CREAR UN ÚNICO OBJETO, YA QUE LAS CLASES ANÓNIMAS NO PUEDEN DEFINIR CONSTRUCTORES.

# FORMAS DE DEFINIR UNA CLASE ANÓNIMA:

## Primera forma:

Las clases anónimas requieren una extensión de la palabra clave new. Se definen en una expresión de Java, incluida en una asignación o en la llamada a un método.

## Segunda forma:

es mediante la palabra new seguida del nombre de la clase de la que hereda (sin extends) y la definición de la clase anónima entre llaves {...}.

## Tercera forma:

es con la palabra new seguida del nombre de la interface que implementa (sin implements) y la definición de la clase anónima entre llaves {...}.

# RELACIONES ENTRE OBJETOS

## LA ASOCIACIÓN

La relación de asociación expresa una relación (unidireccional o bidireccional) entre las instancias a partir de las clases conectadas. El sentido en que se recorre la asociación se denomina navegabilidad de la asociación. Cada extremo de la asociación se caracteriza por el rol o papel que juega en dicha relación el objeto situado en cada extremo.

# RELACIONES ENTRE OBJETOS

## LA COMPOSICIÓN

- La composición consiste en crear una clase nueva agrupando objetos de clases que ya existen. Una composición agrupa uno o más objetos para construir una clase, de manera que las instancias de esta nueva clase contienen uno o más objetos de otras clases.
- Normalmente, los atributos contenidos se declaran con acceso privado (`private`) y se inicializan en el constructor de la nueva clase.

**GRACIAS**

Por su atención



**TECNOLÓGICO  
NACIONAL DE MÉXICO**