


# Clases anidadas



# Definición y tipos

- Se trata de clases que se definen dentro de otras clases
  - Nos podemos encontrar con cuatro situaciones:
    - Clases estándares
    - Clases estáticas
    - Clases locales a método
    - Clases anónimas
- 

# Clases estándares

- Se definen como un miembro más de otra clase:

```
class Externa{  
    class Interna{  
        //metodos y atributos  
    }  
}
```

- Para instanciar una clase interna se necesita un objeto de la externa:

```
Externa ext=new Externa();  
Externa.Interna inter=ext.new Interna();
```

- Una clase interna puede definirse con cualquier modificador de acceso, incluso private

# Clases estáticas

- Se definen como un miembro estático de la clase externa:

```
class Externa{  
    static class Interna{  
        //metodos y atributos  
    }  
}
```

- Para instanciar una clase interna estática no se necesita un objeto de la externa:

```
Externa.Interna inter=new Externa.Interna();
```

- La clase estática interna, solo puede acceder a otros miembros estáticos de la clase Externa

# Clases locales a método

➤ Se definen dentro de un método de la clase externa:

```
class Externa{  
    void metodo(){  
        class Local{  
            //metodos y atributos  
        }  
    }  
}
```

➤ Únicamente pueden ser instanciadas desde el interior del método al que pertenecen, y no tienen acceso a otras variables locales del método, salvo que sean finales.

# Clases anónimas

- Son clases que no tienen nombre. Heredan una clase existente o implementan una interfaz
- Se definen al vuelo, en la misma instrucción de instanciación:

```
Runnable rb=new Runnable(){  
    public void run(){  
        //código del método run  
    }  
}
```

Ejemplo de clase anónima  
que implementa la  
interfaz Runnable

# Revisión conceptos



Dada la siguiente clase, indica que ocurrirá al ejecutar el código que a parece a continuación:

```
class Externa{  
    private static int k;  
    private static class Interna{  
        public void imprimir(){  
            System.out.println(k);  
        }  
    }  
}
```

```
Externa.Interna inter=new Externa.Interna();  
inter.imprimir();
```

**Respuesta**

Se produce un **error de compilación** al intentar crear una instancia de la clase interna, ya que es privada a su clase y no visible desde fuera. Si se hubiera declarado como pública, se mostraría 0

# Revisión conceptos

Discute cuales de las instrucciones marcadas no compila y razonalo:

```
public class InternaMetodo{  
    int a=3;  
    void metodo(){  
        var c=10;  
        c++;  
        class Local{  
            private void met() {  
                a++; //1  
                System.out.println(c); //2  
            }  
        }  
        Local l=new Local(); //3  
    }  
}
```

**Respuesta**

La única que no compila es la 2. Dado que solo es posible acceder a la variable local si es final o efectivamente final, al haber realizado una operación de incremento con la misma se considera variable y no constante