

---

---

# Clase 6: Herencia

— Programación y Laboratorio III —

---

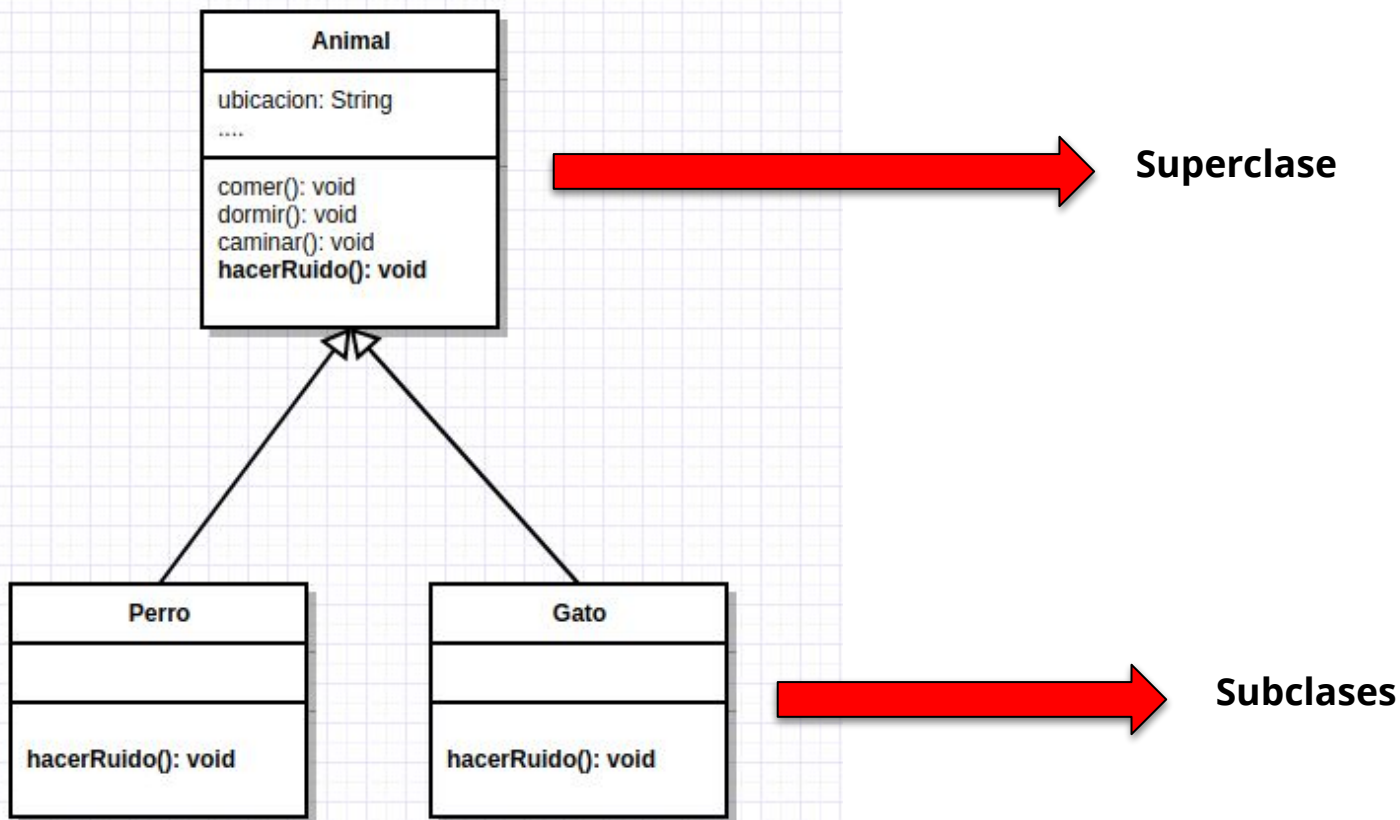
---

# Agenda

- **Herencia**
  - **UML**
  - Definición
  - Superclase y subclases
  - Ventajas
- Herencia y Abstracción
- Herencia en Java
- Clase abstracta
  - Definición
  - Método abstracto
  - Constructores



# Herencia UML - Ejemplo práctico



# Agenda

- **Herencia**

  - UML

  - **Definición**

    - Superclase y subclases
    - Ventajas

- Herencia y Abstracción

- Herencia en Java

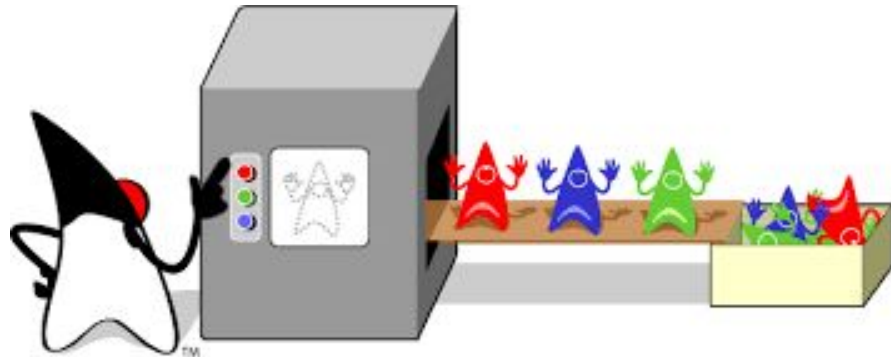
- Clase abstracta

  - Definición
  - Método abstracto
  - Constructores



# Herencia - Definición

- La herencia es un mecanismo que permite la definición de una clase a partir de la definición de otra ya existente.
- Permite compartir automáticamente métodos y características entre clases.
- Está fuertemente ligada a la reutilización de código.



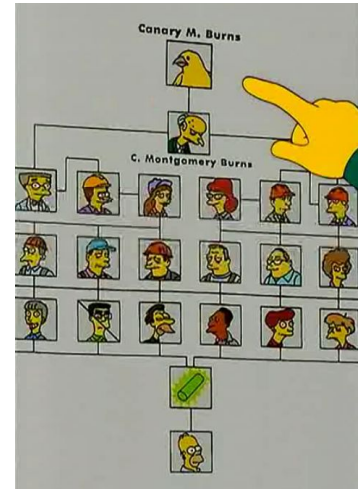
# Agenda

- **Herencia**
  - UML
  - Definición
  - **Superclase y subclases**
  - Ventajas
- Herencia y Abstracción
- Herencia en Java
- Clase abstracta
  - Definición
  - Método abstracto
  - Constructores



# Herencia - Superclase y subclases

- El concepto de herencia conduce a una estructura jerárquica de clases o estructura de árbol.
- En la estructura jerárquica, cada clase tiene sólo una clase padre → **SUPERCLASE**.
  - Una superclase puede tener cualquier número de subclases.
- La clase hija de una superclase se llama **SUBCLASE**.
  - Una subclase puede tener sólo una superclase.
- En el ejemplo:
  - Animal es la superclase de Gato y Perro.
  - Gato y Perro son subclases de Animal.



# Agenda

- **Herencia**
  - UML
  - Definición
  - Superclase y subclases
  - **Ventajas**
- Herencia y Abstracción
- Herencia en Java
- Clase abstracta
  - Definición
  - Método abstracto
  - Constructores





# Herencia - Ventajas

- Evitar la duplicidad y favorecer la **reutilización** de código (las subclases utilizan el código de la superclase).
- Facilitar el **mantenimiento** de las aplicaciones que diseñamos.
- Facilitar la **extensión** de las aplicaciones que diseñamos.

# Agenda

- Herencia
  - UML
  - Definición
  - Superclase y subclases
  - Ventajas
- **Herencia y Abstracción**
- Herencia en Java
- Clase abstracta
  - Definición
  - Método abstracto
  - Constructores



# Herencia y Abstracción

- Se puede pensar en una jerarquía de clases como la definición de **conceptos abstractos** en lo alto de la jerarquía.
- Las subclases no están limitadas al estado y comportamiento provisto por la superclase → pueden agregar variables y métodos además de los que ya heredan.
- Las clases hijas también pueden **sobreescribir** los métodos que heredan.

# Agenda

- Herencia
  - UML
  - Definición
  - Superclase y subclases
  - Ventajas
- Herencia y Abstracción
  - **Herencia en Java**
  - Clase abstracta
    - Definición
    - Método abstracto
    - Constructores

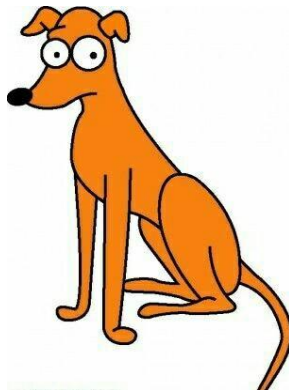


# Herencia en Java

```
public abstract class Animal {  
  
    private String ubicacion;  
    ...  
  
    public abstract void hacerRuido();  
  
    public void comer() {  
        System.out.println("Soy un animal y estoy comiendo");  
    }  
    ...  
}
```

# Herencia en Java (2)

```
public class Perro extends Animal {  
  
    @Override  
    public void hacerRuido() {  
        System.out.println("Guau!!");  
    }  
}
```



```
public class Gato extends Animal {  
  
    @Override  
    public void hacerRuido() {  
        System.out.println("Miau!!");  
    }  
}
```



# Herencia en Java (3)

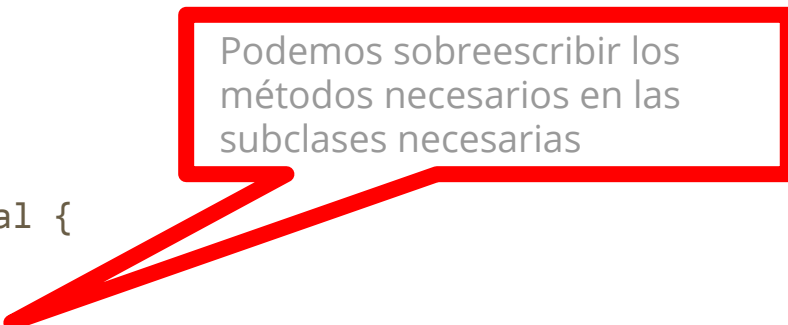
- También se puede heredar **sin tener métodos abstractos**.

```
public class Animal {  
  
    private String ubicacion;  
    ...  
  
    public void comer() {  
        System.out.println("Soy un animal y estoy comiendo");  
    }  
    ...  
}
```

# Herencia en Java (4)

```
public class Perro extends Animal {  
    ...  
}
```

```
public class Gato extends Animal {  
    @Override  
    public void comer() {  
        System.out.println("Soy un gato y estoy comiendo");  
    }  
}
```



Podemos sobrescribir los métodos necesarios en las subclases necesarias



# Agenda

- Herencia
  - UML
  - Definición
  - Superclase y subclases
  - Ventajas
- Herencia y Abstracción
- Herencia en Java
- **Clase abstracta**
  - **Definición**
  - Método abstracto
  - Constructores



# Clase abstracta - Definición

- Es similar a una clase concreta: posee atributos y métodos pero tiene una condición:

- **Al menos uno de sus métodos debe ser abstracto.**

# Agenda

- Herencia
  - UML
  - Definición
  - Superclase y subclases
  - Ventajas
- Herencia y Abstracción
- Herencia en Java
- **Clase abstracta**
  - Definición
  - **Método abstracto**
  - Constructores



# Clase abstracta - Método abstracto

- Un método abstracto se caracteriza por dos detalles:
  - Está precedido por la palabra clave **abstract**.
  - No tiene cuerpo y su encabezado termina con punto y coma.
- Si un método se declara como abstracto, se debe marcar la clase como abstracta → No puede haber métodos abstractos en una clase concreta.
- Los métodos abstractos **deben implementarse** en las clases concretas (subclases). → **@Override**



# Agenda

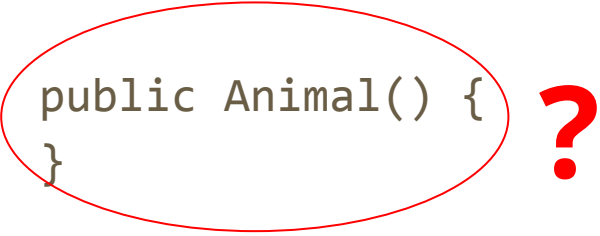
- Herencia
  - UML
  - Definición
  - Superclase y subclases
  - Ventajas
- Herencia y Abstracción
- Herencia en Java
- **Clase abstracta**
  - Definición
  - Método abstracto
  - **Constructores**



# Clase abstracta - Constructores?

- Las clases abstractas no se pueden instanciar!!!

```
public abstract class Animal {  
    ...  
    public Animal() {  
    }  
    public abstract void hacerRuido();  
    ...  
}
```



# Clase abstracta - Constructores? (2)

```
Animal animal = new Animal();
```



- Es posible definir constructores en las superclases, pero no es posible crear instancias.

# Clase abstracta - Constructores? (3)

```
public class Perro extends Animal {
```

```
    private String nombre;
```

```
    public Perro() {
```

```
        super();
```

```
    }
```

```
    public Perro(String nombre) {
```

```
        super();
```

```
        this.nombre = nombre;
```

```
    }
```

```
    ...
```

```
}
```



# Agenda

- Herencia
  - UML
  - Definición
  - Superclase y subclases
  - Ventajas
- Herencia y Abstracción
- Herencia en Java
- Clase abstracta
  - Definición
  - Método abstracto
  - Constructores



# Bibliografía oficial

- <https://docs.oracle.com/javase/tutorial/java/landl/subclasses.html>
- <https://docs.oracle.com/javase/tutorial/java/landl/abstract.html>