



P00: Herencia



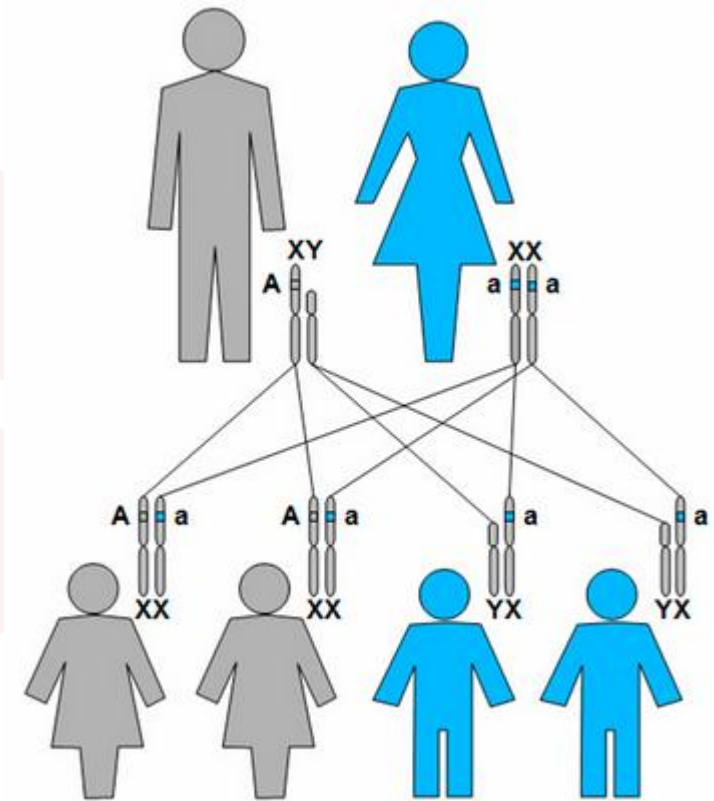
Centro de
Comercio y Turismo



Concepto de Herencia

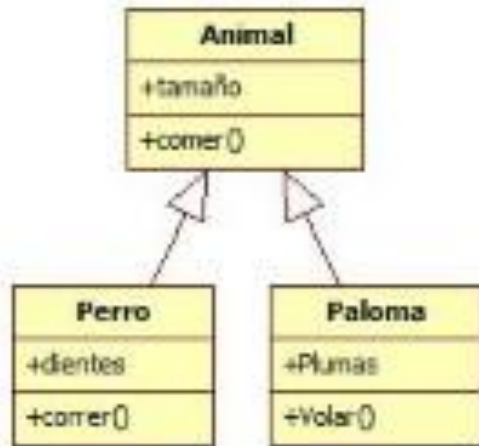
La herencia en java representa lo que conocemos de herencia en el mundo real, básicamente mediante esta obtenemos las características o rasgos comunes de nuestros padres o abuelos.

En java es el mismo enfoque permitiendo la creación de nuevas clases basadas en clases ya existentes, con las cuales podemos obtener las características de las clases padres, heredando campos, atributos, métodos o funcionalidades.



Concepto de Herencia

En Java solo se puede heredar de una sola clase padre y se representa mediante la palabra **extends**



```
public class Animal{
    public String tamaño;

    public void comer(){
        /**Comportamiento.....*/
    }
}
```

```
class Perro extends Animal{
    public int dientes;

    public void correr(){
        /**Comportamiento.....*/
    }
}
```

```
class Paloma extends Animal{
    public int plumas;

    public void volar(){
        /**Comportamiento.....*/
    }
}
```

Concepto de Herencia

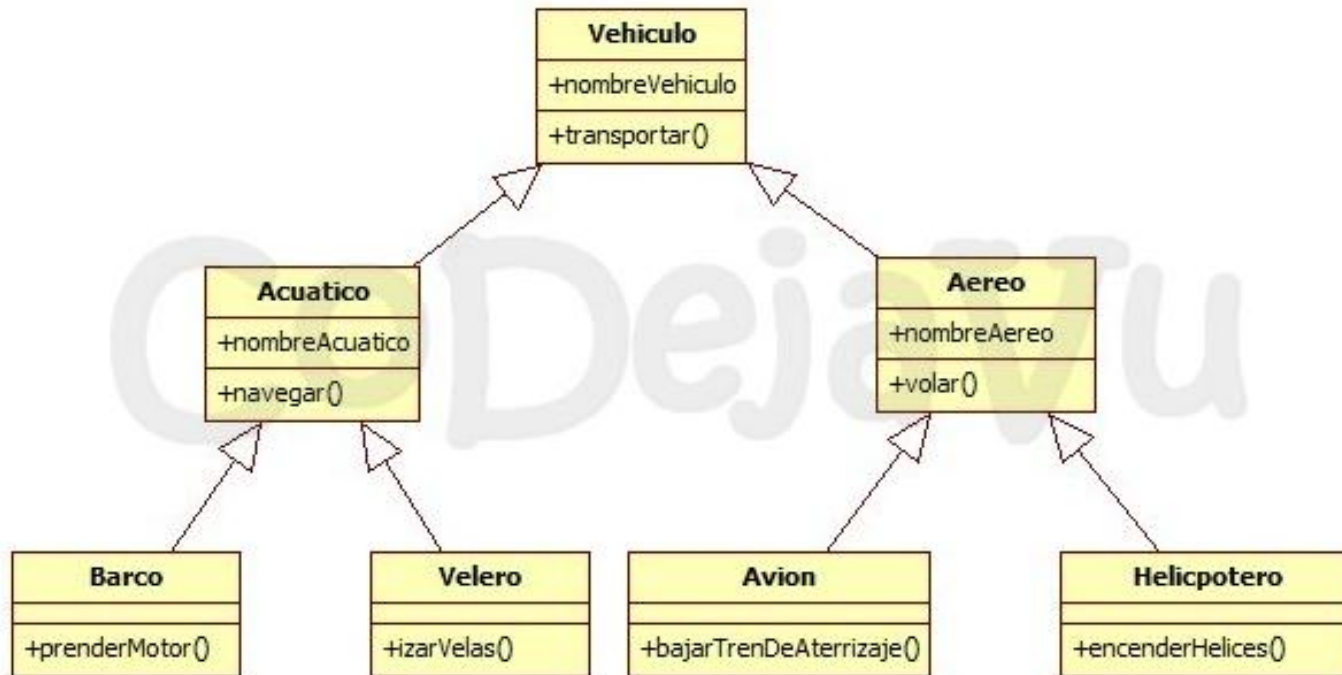
Mediante este concepto podemos obtener las características de una clase padre o de una jerarquía mayor dependiendo de nuestro árbol de herencia.

Como se vio, en java se define la jerarquía de clases mediante la palabra **extends** seguido del nombre de la clase a la que vamos a heredar.

En Java solo se puede heredar de una sola clase pues el lenguaje no permite la Herencia múltiple (En Java por defecto todas las clases heredan de la clase **Object**, es decir, esta es la clase Padre de todas las clases en Java).

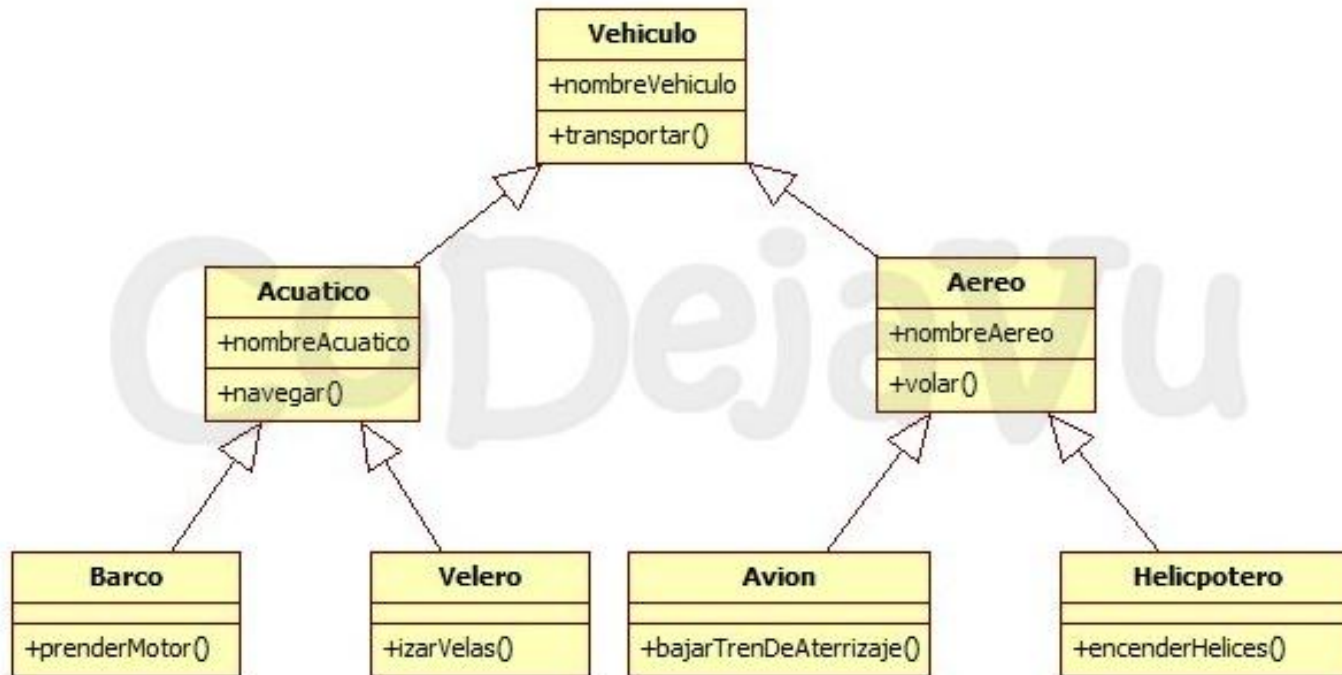


Concepto de Herencia



La Clase **Padre** es **Vehiculo**, la cual tiene como **hijas** a las clases **Acuatico** y **Aereo**, estas a su vez son clases **Padre** de **Barco**, **Velero**, **Avion** y **Helicoptero** que a la vez son **nietas** de **Vehiculo**

Concepto de Herencia



La Clase **Avion** puede usar el método **volar()** de la clase **Aereo**, así como también usar el método **transportar()** de la clase **Vehiculo**, pues se encuentra en el mismo árbol de herencia, sin embargo no puede usar el método **navegar()** de la clase **Acuatico**, ya que no es de tipo **Acuatico**.

Concepto de Herencia

Vehiculo.

Clase Padre.

```

1  /**
2   * @author HENAO
3   *
4   */
5  public class Vehiculo {
6
7      public int modeloVehiculo;
8
9      public String nombreVehiculo="";
10
11     public String transportar(){
12         return "Metodo transportar de clase Vehiculo";
13     }
14 }
```

Acuatico.

SubClase extiende de Vehiculo.

```

1  public class Acuatico extends Vehiculo{
2
3      public String nombreAcuatico="";
4
5      public String navegar(){
6          return "Método navegar de clase Acuatico";
7      }
8
9  }
```

Aereo.

SubClase extiende de Vehiculo.

```

1  public class Aereo extends Vehiculo {
2
3      public String nombreAereo="";
4
5      public String volar(){
6          return "Método volar desde clase Aereo";
7      }
8  }
```

Concepto de Herencia

Acuatico.

SubClase extiende de Vehiculo.

```
1 public class Acuatico extends Vehiculo{
2
3     public String nombreAcuatico="";
4
5     public String navegar(){
6         return "Método navegar de clase Acuatico";
7     }
8
9 }
```

Barco.

SubClase extiende de Acuatico.

```
1 public class Barco extends Acuatico {
2
3     public String prenderMotor(){
4         return "Método prenderMotor en clase Barco";
5     }
6 }
```

Velero.

SubClase extiende de Acuatico.

```
1 public class Velero extends Barco{
2
3     public String izarVelas(){
4         return "Método izarVelas en clase Velero";
5     }
6 }
```


Concepto de Herencia

Aereo.

SubClase extiende de Vehiculo.

```
1 public class Aereo extends Vehiculo {  
2  
3     public String nombreAereo="";  
4  
5     public String volar(){  
6         return "Método volar desde clase Aereo";  
7     }  
8 }
```

Clase Avion.

SubClase extiende de Aereo.

```
1 public class Avion extends Aereo{  
2  
3     public String bajarTrenDeAterrizaje(){  
4         return "Método bajarTrenDeAterrizaje en clase Avion";  
5     }  
6 }
```

Clase Helicoptero.

SubClase extiende de Aereo.

```
1 public class Helicoptero extends Aereo{  
2  
3     public String encenderHelices(){  
4         return "Método encenderHelices en clase Helicoptero";  
5     }  
6 }
```

Concepto de Herencia

```
Acuatico miAcuatico= new Acuatico();  
miAcuatico.nombreVehiculo= "El Navegante";  
System.out.println("usando miAcuatico, nombreVehiculo : "+miAcuatico.nombreVehiculo);  
System.out.println("usando miAcuatico llama a : "+miAcuatico.transportar());  
System.out.println("usando miAcuatico llama a : "+miAcuatico.navegar());  
System.out.println("-----");  
System.out.println();
```

- modeloVehiculo : int - Vehiculo
- nombreAcuatico : String - Acuatico
- nombreVehiculo : String - Vehiculo
- hashCode() : int - Object
- navegar() : String - Acuatico
- toString() : String - Object
- transportar() : String - Vehiculo
- equals(Object obj) : boolean - Object
- getClass() : Class<?> - Object
- notify() : void - Object
- notifyAll() : void - Object
- wait() : void - Object

Press 'Ctrl+Space' to show Template Proposals

Concepto de Herencia

La herencia va ligada a muchos otros conceptos, se aplica el concepto de clases ya que la herencia se trabaja con estas.

El concepto de objetos pues la comunicación entre clases hijas y clases padres depende del llamado de métodos entre objetos

Por ultimo se tiene el concepto de encapsulación ya que como compartimos características entre clases tenemos que estar seguros que la clase que accede puede usarlas...



FUENTES

- Introducción a la Programación en Java2 1.5 (Java2 5.0), Universidad del Quindío, Facultad de Ingeniería, Ingeniería de Sistemas y Computación, Ing Julian Esteban Gutierrez
- Fundamentos de programación. Problemas resueltos de programación en lenguaje Java. José María Pérez Menor, Jesús Carretero Pérez, Félix García Carballeira, and José Manuel Pérez Lobato. Madrid: Paraninfo, 2003. p[1]-22
- <http://codejavu.blogspot.com/2013/05/conceptos-de-programacion-orientada.html>
- <http://codejavu.blogspot.com/2013/05/herencia-en-java.html>