

Universidad Autónoma de Baja California
Facultad de Ciencias de la Ingeniería y Tecnología
Materia:LPOO



#3-Herencia Completa en Java con

Alumno: Ángel Alejandro reyes morales

Matricula: 2206952

Grupo: 544

Profesor: Emmanuel Gómez Gutiérrez

Fecha:31/10/2025

Tijuana, Baja California, México

```

ar_impar.java 1  J Listatareas8.java 1  J Practica_del_DRIVE_THRU9.java  J funciones_calculadora10.java 1  J BancoCajero.java 1  J Auto.java
J Veterinaria.java > Mascota
1  import java.util.ArrayList;
2  import java.util.Scanner;
3
4  // Clase abstracta Mascota
5  abstract class Mascota {
6      protected String nombre;
7      protected int edad;
8
9      public Mascota(String nombre, int edad) {
10         this.nombre = nombre;
11         this.edad = edad;
12     }
13
14     public abstract void hablar();
15     public abstract void presentar();
16
17     public String getNombre() {
18         return nombre;
19     }
20
21     public int getEdad() {
22         return edad;
23     }
24
25     public void setNombre(String nombre) {
26         this.nombre = nombre;
27     }
28
29     public void setEdad(int edad) {
30         this.edad = edad;
31     }
32 }
33
34 class Perro extends Mascota {
35     private String raza;
36
37     public Perro(String nombre, int edad, String raza) {
38         super(nombre, edad);
39         this.raza = raza;
40     }
41
42     @Override
43     public void hablar() {
44         System.out.println(nombre + " dice: ¡Guau guau!");
45     }
46
47     @Override
48     public void presentar() {
49         System.out.println(" Perro - Nombre: " + nombre + ", Edad: " + edad + " años" + ", Raza: " + raza);
50     }
51 }

```

```

48     public void presentar() {
49     }
50
51     public String getRaza() {
52         return raza;
53     }
54
55     public void setRaza(String raza) {
56         this.raza = raza;
57     }
58 }
59
60
61 class Gato extends Mascota {
62     private String color;
63
64     public Gato(String nombre, int edad, String color) {
65         super(nombre, edad);
66         this.color = color;
67     }
68
69     @Override
70     public void hablar() {
71         System.out.println(nombre + " dice: ¡Miau miau!");
72     }
73
74     @Override
75     public void presentar() {
76         System.out.println("Gato - Nombre: " + nombre + ", Edad: " + edad + " años" + ", Color: " + color);
77     }
78
79     public String getColor() {
80         return color;
81     }
82
83     public void setColor(String color) {
84         this.color = color;
85     }
86 }
87
88 public class Veterinaria {
89     Run | Debug
90     public static void main(String[] args) {
91         Scanner entrada = new Scanner(System.in);
92         ArrayList<Mascota> mascotas = new ArrayList<>();
93         int opcion = 0;
94
95         while (opcion != 7) {
96             System.out.println(x:"MENÚ VETERINARIA");
97             System.out.println(x:"1. Registrar Perro");
98             System.out.println(x:"2. Registrar Gato");

```

```

89     public static void main(String[] args) {
90         //
91         System.out.println(x:"3. Mostrar Mascotas");
92         System.out.println(x:"4. Hacerlas hablar");
93         System.out.println(x:"5. Buscar Mascota por nombre");
94         System.out.println(x:"6. Eliminar Mascota");
95         System.out.println(x:"7. Salir");
96         System.out.print(s:"Elige una opción: ");
97         opcion = entrada.nextInt();
98         entrada.nextLine(); // limpiar buffer
99
100         if (opcion == 1) {
101             System.out.print(s:"Nombre del perro: ");
102             String nombreP = entrada.nextLine();
103             System.out.print(s:"Edad: ");
104             int edadP = entrada.nextInt();
105             entrada.nextLine();
106             System.out.print(s:"Raza: ");
107             String raza = entrada.nextLine();
108
109             mascotas.add(new Perro(nombreP, edadP, raza));
110             System.out.println(x:"Perro agregado correctamente.");
111
112         } else if (opcion == 2) {
113             System.out.print(s:"Nombre del gato: ");
114             String nombreG = entrada.nextLine();
115             System.out.print(s:"Edad: ");
116             int edadG = entrada.nextInt();
117             entrada.nextLine();
118             System.out.print(s:"Color: ");
119             String color = entrada.nextLine();
120
121             mascotas.add(new Gato(nombreG, edadG, color));
122             System.out.println(x:"Gato agregado correctamente.");
123
124         } else if (opcion == 3) {
125             System.out.println(x:" Mascotas Registradas");
126             if (mascotas.isEmpty()) {
127                 System.out.println(x:"No hay mascotas registradas.");
128             } else {
129                 for (Mascota m : mascotas) {
130                     m.presentar();
131                 }
132             }
133
134         } else if (opcion == 4) {
135             System.out.println(x:"Las mascotas hablan");
136             if (mascotas.isEmpty()) {
137                 System.out.println(x:"No hay mascotas registradas.");
138             } else {
139                 for (Mascota m : mascotas) {

```

```

89     public static void main(String[] args) {
146         for (Mascota m : mascotas) {
147             m.hablar();
148         }
149     }
150
151     } else if (opcion == 5) {
152         System.out.print(s:"Ingrese el nombre de la mascota a buscar: ");
153         String nombreBuscar = entrada.nextLine();
154         boolean encontrada = false;
155
156         for (Mascota m : mascotas) {
157             if (m.nombre.equalsIgnoreCase(nombreBuscar)) {
158                 System.out.println(x:"Mascota encontrada:");
159                 m.presentar();
160                 encontrada = true;
161                 break;
162             }
163         }
164
165         if (!encontrada) {
166             System.out.println(x:"No se encontró ninguna mascota con ese nombre.");
167         }
168
169     } else if (opcion == 6) {
170         System.out.print(s:"Ingrese el nombre de la mascota a eliminar: ");
171         String nombreEliminar = entrada.nextLine();
172         Mascota mascotaEliminar = null;
173
174         for (Mascota m : mascotas) {
175             if (m.nombre.equalsIgnoreCase(nombreEliminar)) {
176                 mascotaEliminar = m;
177                 break;
178             }
179         }
180
181         if (mascotaEliminar != null) {
182             mascotas.remove(mascotaEliminar);
183             System.out.println(x:"Mascota eliminada correctamente.");
184         } else {
185             System.out.println(x:"No se encontró ninguna mascota con ese nombre.");
186         }
187
188     } else if (opcion == 7) {
189         System.out.println(x:"Saliendo del programa...");
190     } else {
191         System.out.println(x:"Opción no válida. Intenta de nuevo.");
192     }
193 }
194

```

```
PS C:\Users\johan\Downloads\LPP0\Ejercicios de java> & 'C:\Program Files\Java\jdk-25\bin\java.exe' '--enable-
\Roaming\Code\User\workspaceStorage\482268d3eccc11e0e22d4fc2e98fd1c\redhat.java\jdt_ws\Ejercicios de java_99d
MENÚ VETERINARIA
1. Registrar Perro
2. Registrar Gato
3. Mostrar Mascotas
4. Hacerlas hablar
5. Buscar Mascota por nombre
6. Eliminar Mascota
7. Salir
Elige una opción: 1
Nombre del perro: LEO
Edad: 3
Raza: pug
Perro agregado correctamente.
MENÚ VETERINARIA
1. Registrar Perro
2. Registrar Gato
3. Mostrar Mascotas
4. Hacerlas hablar
5. Buscar Mascota por nombre
6. Eliminar Mascota
7. Salir
Elige una opción: 3
Mascotas Registradas
Perro - Nombre: LEO, Edad: 3 años, Raza: pug
MENÚ VETERINARIA
1. Registrar Perro
2. Registrar Gato
3. Mostrar Mascotas
4. Hacerlas hablar
5. Buscar Mascota por nombre
6. Eliminar Mascota
7. Salir
Elige una opción: █
```