

Guía Práctica de Laboratorio

Sesión 15: INTRODUCCIÓN A LA PROGRAMACIÓN

MLBC - VACJ - CJFV

En esta sesión se trabajará por un lado en la capacidad de comprensión de código utilizando herencia y las ventajas que ella ofrece

Por otro el planteamiento de una propuesta de solución desde cero, utilizando herencia.

1. Ping Pong

Como ya sabes Ping Pong es un muñeco muy guapo y de cartón, así como él existen muchos muñecos muy guapos y de cartón que realizan distintas labores cotidianas. Estos muñecos puede trabajar haciendo varias cosas y dependiendo de su trabajo, se lavan la cara con agua y algún detergente especial a su rubro. Por ejemplo si hace tareas de escritorio se lava con agua y jabón, si realiza tareas involucradas con la medicina se lava con agua y desinfectante, por último, si realiza tareas artísticas se lava la cara con agua y desmaquillante. Cada uno de estos muñecos además tiene un nombre.

Considerando este contexto se ha planeado el modelo de la figura 1

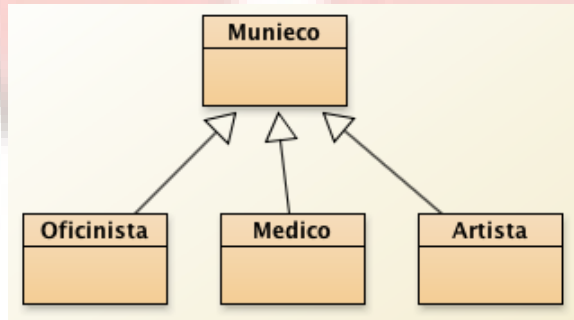


Figura 1: Modelo del problema

Y el código de este modelo es:

```
1 class Muneco{
2     protected String nombre;
3     public String cantar(){
4         String cancion;
5         cancion = nombre;
6         cancion = cancion + "es un muñeco muy guapo y de carton , se lava la carita con agua y ";
7         return cancion;
8     }
9 }
10 class Artista extends Muneco{
11     public Artista(String nombre){
12         this.nombre = nombre;
13     }
14     public String cantar(){
15         String cancion;
16         cancion = nombre;
17         cancion = cancion + "es un muñeco muy guapo y de carton , se lava la carita con agua y desmaquillante";
18         return cancion;
19     }
20 }
21 class Oficinista extends Muneco{
22     public Oficinista(String nombre){
23         this.nombre = nombre;
24     }
25     public String cantar(){
26         String cancion;
27         cancion = nombre;
28         cancion = cancion + "es un muñeco muy guapo y de carton , se lava la carita con agua y jabon";
```

```

29         return cancion;
30     }
31 }
32 class Medico extends Munieco{
33     public Medico(String nombre){
34         this.nombre = nombre;
35     }
36     public String cantar(){
37         String cancion;
38         cancion = nombre;
39         cancion = cancion + " _es _un _munieco _muy _guapo _y _de _carton , _se _lava _la _carita _con _agua _
            y _desinfectante";
40         return cancion;
41     }
42 }

```

De tal manera que si se tiene un objeto de clase Munieco cuyo nombre es "Pedrito" que es un artista y le pedimos que cante, él cantará así: "Pedrito es un munieco muy guapo de carton, se lava la carita con agua y desmaquillante"

Es mas, si se tuviera el escenario descrito en el siguiente código:

```

1 Munieco m;
2 String cancion;
3 m = new Artista("Pedrito");
4 cancion = m.cantar();
5 System.out.println(cancion);
6 m = new Oficinista("Juanito");
7 cancion = m.cantar();
8 System.out.println(cancion);
9 m = new Medico("Pablito");
10 cancion = m.cantar();
11 System.out.println(cancion);

```

En línea de comandos se mostrará el contenido de la cadena canción, en tres oportunidades, después de ejecutar la:

línea 5 "Pedrito es un munieco muy guapo de carton, se lava la carita con agua y desmaquillante"

línea 8 "Juanito es un munieco muy guapo de carton, se lava la carita con agua y jabon"

línea 11 "Pablito es un munieco muy guapo de carton, se lava la carita con agua y desinfectante"

Para probar esto puedes usar el codepad de BlueJ.

Tu tarea es mejorar la implementación presentada de tal manera de reusar lo mas que se pueda el código.

Una vez que mejores el código, vuelve a ejecutar el escenario de prueba y verifica que el objeto se comporte de la misma manera que antes. Explica por que sucede esto, para ello puedes utilizar el archivo README de tu proyecto BlueJ

2. Caracol Caracolito

Los caracoles son animales que se arrastran para moverse de un lado a otro, en esta ocasión se planea tener dos tipos de caracoles que a medida que se arrastran, arrasan con todo lo que tienen en su paso, unos se mantienen en una línea y suelen ser flaquito, en cambio hay otros que a medida que se mueven van sumándole materia a su cuerpo y a cada paso que dan arrasan mas de un elemento en su camino.

a) Caracol flaquito

0	1	2	3	4	5	6	7	8	9	10	11
4	6	10	13	5	8	2	1	5	7	9	1

El caracol arrasa un total de 33 en valor

b) Caracol gordito

	0	1	2	3	4	5	6	7	8	9	10	11
0	2	4	9	11	3	4	18	2	4	1	2	16
1	3	6	7	10	17	9	2	5	2	16	4	5
2	2	8	8	1	5	2	7	9	7	9	10	23
3	2	9	3	4	2	6	8	8	4	15	3	7
4	3	3	4	5	6	1	8	7	9	9	2	3
5	5	2	4	3	4	3	11	4	2	1	10	4
6	2	12	13	10	3	8	10	5	6	3	8	5
7	3	10	17	9	2	12	13	2	1	14	1	17
8	5	7	6	6	3	7	7	5	3	9	4	7
9	11	8	2	2	7	3	5	9	8	15	2	3
10	3	6	7	10	17	9	1	8	12	1	2	7
11	5	2	5	11	1	6	12	3	11	4	5	3
12	1	2	5	9	10	8	23	54	32	10	9	8
13	7	3	2	10	2	7	9	10	5	14	9	18
14	11	5	7	3	4	5	23	2	3	1	22	6
15	9	11	8	10		9	3	5	6	7	8	3
16	13	7	8	8	12	1	2	13	2	20	5	19
17	3	6	7	10	17	9	3	5	6	7	8	4

El caracol arrasa un total de 351 en valor

Figura 2: Ejemplo de Caracolos

Para fines del ejercicio, es correcto decir que los caracolitos flaquitos se mueven siempre en un arreglo de elementos numéricos y el segundo tipo de caracolito que son los gorditos, se mueven en un espacio bidimensional de elementos numéricos. Ambos parten de una posición y avanzan hacia “adelante”, de la forma que se muestra en la figura 2

En el ejemplo de la figura 2, al finalizar el avance un caracolito flaquito, que haya iniciado su caminata en la posición 5, habrá arrasado con los valores {8, 2, 1, 5, 7, 9, 1} del plano sobre el cual camina.

En cambio si se tiene un caracolito gordito que inicia su caminata en la posición (3,4) habrá arrasado con los valores {2, 1, 8, 3, 11, 4, 2, 10, 13, 5, 6, 3, 8, 2, 5, 9, 1, 14, 1, 17, 3, 8, 12, 11, 9, 4, 7, 15, 1, 4, 10, 14, 2, 3, 2, 5, 9, 9, 22, 8, 7, 3, 8, 18, 6, 3, 19, 4} del plano sobre el cual camina. Al finalizar los caracolos deben calcular la suma de los valores que arrasa; el flaquito en el ejemplo recoge una suma de 33 y el gordito en el ejemplo recoge 351.

Debes cuidar que el plano sobre el cual pones a caminar un caracolito debe coincidir y ser consistente con el caracolito. (hay dos tipos de planos...)

Sobre la base de la descripción del problema, se pide:

- modelar la solución al problema
- implementa lo necesario para permitir a un caracolito caminar sobre un espacio consistente con su tipo, al terminar su caminata debes informar cuanto de valor arraso.