

RESUMEN PARADIGMA ORIENTADO A OBJETOS
FACULTAD REGIONAL BUENOS AIRES
UNIVERSIDAD TECNOLÓGICA NACIONAL

PARADIGMAS DE PROGRAMACIÓN – *K2032*

Trabajos Prácticos

PAZ PORTILLA, José Miguel 2028244
jpazportilla@frba.utn.edu.ar

15 de noviembre de 2023

Índice

1. Introducción a Objetivo

En este paradigma de objetos se vuelve a tener efecto, pero es menos declarativo que funcional y lógico. Se utiliza el lenguaje de programación Wollok. La idea es combinar estructuras de datos y operaciones.

Las características de un objeto son:

1. Exponen una interfaz, es un conjunto de operaciones con las que se pueden interactuar con el objetos. Solo se puede interactuar con objetos mediante mensajes. Los mensajes que un objeto entiende va a ser el resultado de poseer metodos.
2. Pueden llegar a tener estado interno, que son atributos, es decir referencias a otros objetos. Estos atributos pueden cambiar de referencia y apuntar a otros objetos.
3. Tienen una identidad, cada objeto es diferente a cualquier otro, aunque hayan otros que respondan a los mismos mensajes y estado interno.

1.1. Problema: La Golondrina Pepita

- Un ornitólogo nos pide ayuda para estudiar el Consumo de Energía de la golondrina Pepita
- El Volar consume energia de Pepita.
- El Comer recupera la energía de Pepita.

1.1.1. Resolución: La Golondrina Pepita

- La palabra `object` define un objeto nuevo.
- La palabra `var` define un atributo que podrá ser cambiado
- La palabra `method` permite crear métodos.
- El metodo `volar()` y `comer()` causan efecto.
- El metodo `energia()` son solo de consulta.

1.1.2. Wollok: La Golondrina Pepita

```
1 object pepita
2 {
3     var energia = 100
4
5     method vola (kilometros)
6     {
7         energia = energia - kilometros * 2
8     }
9
10    method come (gramos)
11    {
12        energia = energia + gramos * 10
13    }
14
15    //Es un getter, obtiene el valor del atributo energia y lo retorna
```

```

16 //Cuando retorna una expresion se utiliza esa forma de definir metodos
17 method energia () = energia
18 //Se podria haber definido asi
19 /*
20     method energia ()
21     {
22         return energia
23     }
24 */
25 }

```

1.2. Problema: La entrenadora de aves Emilia

- Emilia solo sabe entrenar aves
- La aves deben comer 5g, volar 10km y volver a comer 5g.

1.2.1. Resolución: La entrenadora de aves Emilia

- Emilia no conoce a Pepita, pero como Pepita entiende los mensajes come y vola, entonces podra entrenarla.

1.2.2. Wollok: La entrenadora de aves Emilia

```

1 import pepita.*
2 import pepote.*
3
4 object emilia
5 {
6     method entrena (ave)
7     {
8         ave.come(5)
9         ave.vola(10)
10        ave.come(5)
11    }
12 }

```

1.3. Emilia entrena a Pepita

Emilia puede entrenar a cualquier objeto que entienda los mensajes come y vola. En la ?? se observa que pepita sufre el efecto luego que emilia la entrena, aumentando su energia de 100J a 180J.

1.3.1. Wollok: Emilia entrena a Pepita

```

1 import pepita.*
2 import emilia.*
3
4 describe "Emilia conoce a su golondrina Pepita"
5 {
6     test "Pepita inicia con 100J de ejergia"

```

```
7 {  
8     assert.equals(100, pepita.energia())  
9 }  
10 test "Emilia entrena a Pepita y finalmente tiene 180J de energia"  
11 {  
12     emilia.entrena(pepita)  
13     assert.equals(180, pepita.energia())  
14     /*Ya que luego de volar 5km su energia disminuyo en 2*5=10  
15      * Luego comio 10g y su energia aumento en 10*10=100  
16      * Finalmente volvio a volar 5km y su energia disminuto en 2*5=10  
17      * por lo tando la energia final resulta 100-10+100-10=180  
18      */  
19 }  
20 }
```

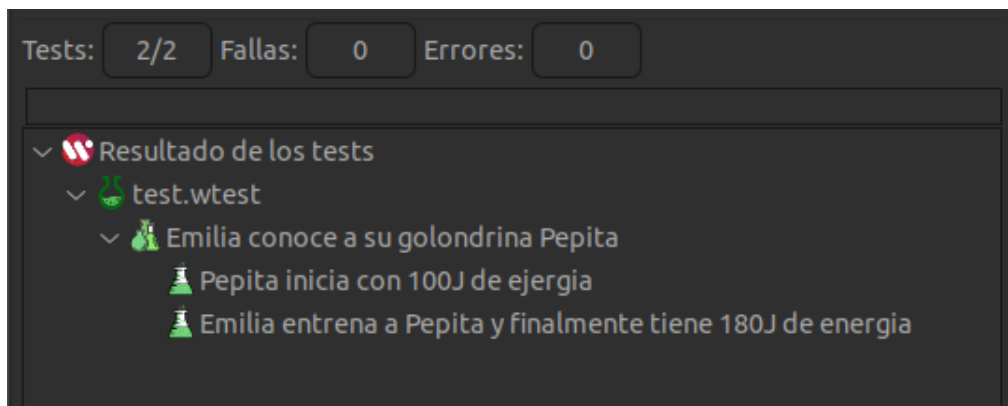


Figura 1. Test unitario de pepita siendo entrenada por emilia.

1.4. Conceptos clave de la programación orientada a objetos

A. Apéndice