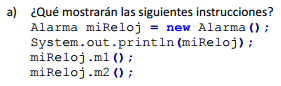
Universidad del Valle de Guatemala

Rodrigo Stuardo Juárez Jui 16073

Laboratorio 4: Herencia

Ejercicio 1: Timbre y Alarma



Las siguientes instrucciones mostrarían:

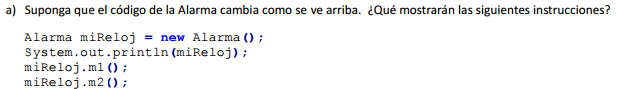
Ring...Ring (toString del método padre)

Alarma1 (método m1 de Alarma)

timbre2 (método m2 de Timbre)

b) ¿Por qué sale eso en pantalla?  
 Sale eso en pantalla debido a que como se llama a la clase hija, esta también tiene acceso, hereda, los métodos de la clase Timbre. Por lo tanto al llamar a miReloj se muestra el toString de la clase Timbre. Al llamar al método m1, como la clase Alarma tiene su propio método m1, este se sobreescribe y se imprime el método de la clase Alarma. El método m2 no es sobreescrito por lo que se llama al método de la clase padre: Timbre.

Ejercicio 2:



Se imprime:

Ring…Ring…Ring…Ring

Alarma 1

timbre 1

b) ¿Por qué sale eso en pantalla?  
La primera línea se imprime debido a que primero se llama al toString de Alarma y este lo que hace es llamar dos veces al toString de la clase Padre y los une mediante “…”. Luego el método m1 está sobreescrito e imprime el método de la hija. El método m2 también está sobreescrito pero este regresa el método m1 de la clase padre.

Ejercicio 3:

Pasos para la solución:

* Crear una nueva clase: AlarmaEmergencia que sea hija de Alarma
* Crear un método m1 que sobreescriba el método de Alarma para que imprima “Emergencia 1”
* Crear un método m2 que llame a los métodos m1 y m2 de Alarma para que estos se impriman ya que estos tienen la instrucción de imprimir “alarma 1” y “timbre 1” respectivamente. “timbre 1” se llama desde la clase Timbre que es padre de Alarma
* Se crea un toString de la clase AlarmaEmergencia en el cual primero se coloca “emergencia “ y se la agrega el método toString de la clase Alarma que incluye el “Ring…Ring…Ring…Ring”

Ejercicio 4:

Pasos:

* Se creó una clase soldado que fuera hija de Militar
* Se hizo un método getSalario que usara el dato del padre y les restara 10000 para tener su salario
* Se creó una metodo getHoras que también usara los datos del padre y los multiplicara por dos
* Se creó un método que usara los datos del padre y los dividiera dentro de 2
* Se creó un método recibeOrden que regresara “Ordene mi General”
* Se comprobó en un Driver que funcionara

Repositorio GitHub

<https://github.com/jua16073/Laboratorio-4>