

Tema 2: Lección 1

Ejercicio 1: Responde verdadero (V) o falso (F)

Dos objetos con el mismo estado pueden tener distinta identidad.	
Si hay mil objetos de una clase X habrá mil copias de su variable de clase x1.	
El código: MuertoViviente vampiro ; crea en Java un objeto de la clase MuertoViviente.	
El código: attr_writer :color crea el consultor y el modificador del atributo color en Ruby.	
En Ruby los métodos de instancia son públicos y los atributos de instancia son privados, por defecto.	
La identidad de un objeto en programación orientada a objetos la da su dirección de memoria.	
Los constructores por defecto devuelven void.	
La encapsulación de un conjunto de elementos implica de forma implícita su ocultamiento para el resto de elementos del sistema.	
Cuando definimos una clase, definimos el estado y el comportamiento de un conjunto de objetos, y en algunas ocasiones también definimos estado y/o comportamiento de la propia clase.	
Todos los lenguajes de programación soportan los siguientes atributos de visibilidad de forma explícita para sus atributos: private , package y public	
Para invocar a los métodos de clase no es necesario que exista previamente una instancia de dicha clase en el sistema.	
Los paquetes son específicos de Java.	
En Java y Ruby las clases se tratan como objetos mientras que en C++ las clases constituyen un patrón.	

Ejercicio 2. Sea la clase java Rectángulo:

```
class Rectangulo {
    float ladoMenor, ladoMayor;
    String color;
    Rectangulo(){
        ladoMenor=2;
        ladoMayor=4;
        color="azul";
    }
    void setColor(String unColor) {
        color = unColor;
    }
    float area(){
        return ladoMenor*ladoMayor;
    }
}
```

Representa lo que va ocurriendo con cada una de estas instrucciones. Se proporciona el resultado de los pasos (1) y (2), haz una figura para cada uno de los pasos (3), (4), (5) y (6).

- (1) Rectangulo a, b, c;
- (2) a = new Rectangulo();
- (3) b = new Rectángulo();
- (4) a.setColor("rojo");

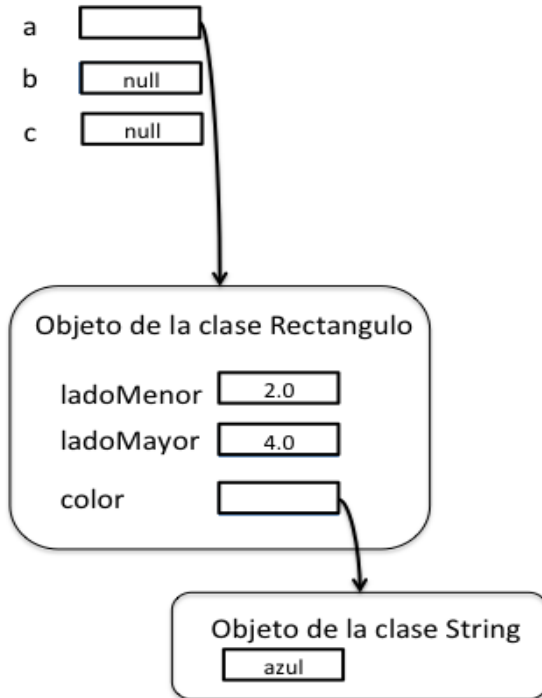
(5) `float x = a.area();`

(6) `c = a;`

Paso (1)



Paso (2)



Ejercicio 3. Dada la clase Java:

```

public class Prueba {
    public static final int A = 1;
    static String s = "";
    private int b = 2;
    int c;
}
  
```

a. ¿Cuántos atributos tiene? Indica si se trata de atributos de instancia o de clase. ¿Cuál es la visibilidad de cada uno de ellos?

b. A partir del siguiente código, suponiendo que está en otra clase del mismo paquete:

1. `Prueba obj1 = new Prueba();`
2. `Prueba obj2 = new Prueba();`
3. `obj1.A = 3;`
4. `Prueba.s = "hola";`
5. `Prueba.A = 14;`
6. `obj1.b = 0;`
7. `obj2.b = 5;`
8. `obj1.c = 4;`
9. `obj2.c = 6;`

b.1. ¿Se produciría algún error de compilación? ¿por qué? (Recomendación: ejecutar el código para comprobarlo).

b.2. Indica cuál es el estado de obj1 y obj2 después de su ejecución y eliminados los errores de compilación si los había.

Ejercicio 4. ¿Quién tiene la responsabilidad de responder a los mensajes que se corresponden con los métodos de clase?

Ejercicio 5. Razona si las siguientes afirmaciones son ciertas o falsas, suponiendo que estamos en la clase citada:

1. Los atributos de clase son accesibles sólo desde métodos de clase (no desde métodos de instancia)
2. Los atributos de instancia son accesibles sólo desde métodos de instancia (no desde métodos de clase).
3. La palabra reservada "this" (Java) puede emplearse tanto en métodos de clase como de instancia.

Ejercicio 6. ¿Qué mecanismos tiene Java para la ocultación de información? Y ¿Ruby?

Ejercicio 7.

Sea la siguiente clase en Java:

```
Class C {  
    private static int contador2 = 0;  
    private int contador1= 0;  
    public void incrementarContador1() {contador1++;}  
    public void incrementarContador2() {contador2++;}  
    public int getContador1() {return contador1;}  
    public int getContador2() {return contador2;}  
}
```

Tras ejecutar el siguiente trozo de código:

```
C objeto1 = new C();  
C objeto2 = new C();  
objeto1.incrementaContador1();  
objeto1.incrementaContador2();  
objeto2.incrementaContador1();  
objeto2.incrementaContador2();  
int valor1 = objeto1.getContador1();  
int valor2 = objeto1.getContador2();  
int valor3 = objeto2.getContador1();  
int valor4 = objeto2.getContador2();
```

1. ¿Qué valores tienen valor1, valor2, valor3 y valor4?

2. Traduce el código Java proporcionado a Ruby

Ejercicio 8. varibales de instancia en Ruby

Sea el siguiente código en Ruby:

```
class A  
    @variable_instancia = 50  
    def variable_instancia  
        puts "valor de la variable de instancia: #{@variable_instancia}"  
    end  
end
```

```

end
def self.variable_instancia
  puts "valor de la variable de instancia: #{@variable_instancia}"
end
end

```

Indica cuál es el resultado de ejecutar las siguientes líneas de código

A.new.variable_instancia

A.variable_instancia

Explica el porqué de estos resultados

Ejercicio 9. Tipo de variables en Ruby

Sea el siguiente código en Ruby:

```

class A
  CONSTANT= 0
  $Global= 10
  @@atributo_clase=20

  def initialize
    @atributo_instancia=40
  end
  def atributo_instancia
    puts "valor del atributo de instancia #{@atributo_instancia}"
  end
  def atributo_clase
    puts "valor del atributo de clase #{@@atributo_clase}"
  end
  def self.atributo_clase
    puts "valor del atributo de clase #{@@atributo_clase}"
  end
end

```

Indica cuál es el resultado de ejecutar las siguientes líneas de código

1. puts A::CONSTANT
2. A::CONSTANT = 2
3. puts A::CONSTANT

4. puts "valor de la global #{\$Global}"
5. \$Global = 30
6. puts "valor de la global #{\$Global}"

7. A.new.atributo_instancia

8. A.new.atributo_clase

9. A.atributo_clase

Explica el porqué de estos resultados

Ejercicio 10. Paquetes y módulos

Sean los ficheros A.java y B.java cuyo contenido es:

A.java

```
package p1;
import p2.B;
class A { (...) }
```

B.java

```
package p2;
public class B {
    String atributo1;
    public String atributo2;
    (...) }
```

Responde Verdadero (V) o Falso (F)

1. La clase A pertenece al paquete p1

2. Como se ha hecho un import de la clase B, atributo1 es accesible desde A

3. Como se ha hecho un import de la clase B, atributo2 es accesible desde A

Tra ¿Es posible traducir exactamente a Ruby el código anterior? En caso negativo, ¿cuál sería la forma más próxima de implementar una estructura similar?

Ejercicio 11. Acceso a métodos en Ruby

Sea el siguiente código en Ruby:

```
class Ejemplo
  def publico_implicito_protegido
    metodo_protegido
  end
  def publico_explicito_protegido
    self.metodo_protegido
  end
  def publico_implicito_privado
    metodo_privado
  end
  def publico_explicito_privado
    self.metodo_privado
  end

  protected
  def metodo_protegido
    puts "metodo protegido"
  end
  private
  def metodo_privado
    puts "método privado"
  end
end
```

Indica cuál es el resultado de ejecutar las siguientes líneas de código	
1. Ejemplo.new.publico_implicito_protegido	
2. Ejemplo.new.publico_explicito_protegido	
3. Ejemplo.new.metodo_protegido	
4. Ejemplo.new.publico_implicito_privado	
5. Ejemplo.new.publico_explicito_privado	
6. Ejemplo.new.metodo_privado	
Explica el resultado obtenido en cada uno de estos puntos	

Ejercicio 12. Las siguientes instrucciones en Java pretenden declarar Arrays. Haz pruebas en Java para entender cómo funcionan. Indica qué errores ves y escríbelas correctamente en Java.

	Motivo de error, si lo hay	Solución al error, en su caso
int [][] c;		
int [5] e;		
int d[];		
int f[]=new int[3];		
int[] x=new int[10];		
Array.newInstance(int, 5);		
int [] dims={2,4};		
Array.newInstance(Alumno, dims);		

Ejercicio 13. Define en Java y en Ruby una clase cuyas instancias representen atletas y otra clase cuyas instancias sean un equipo de atletas y un entrenador al frente. Incluye los atributos que consideres necesarios. Además, ten en cuenta que necesitamos saber cuántos equipos de atletas hay. Escribe un programa sencillo que cree un equipo y muestre

- Los atletas que corren en el mismo y su entrenador.
- El número de equipos que tenemos

Ejercicio 14. Tipo de variables en Ruby

Sea el siguiente código en Ruby:

```
class Planeta
  NOMBRES=['Mercurio','Venus','Tierra','Marte','Júpiter',
          'Saturno','Urano','Neptuno'].freeze
  @@planetas_instanciados = 0

  def initialize
    if @@planetas_instanciados >= NOMBRES.size
      raise ArgumentError, 'Lo siento, pero ya se han creados todos
                          los planetas, ahora a descansar'
    end
  end
end
```

<pre>@nombre = NOMBRES[@@planetas_instanciados] @@planetas_instanciados += 1 puts "Se ha creado el planeta #{@nombre}!" end</pre>	
end	
Indica cuál es el resultado de ejecutar la siguiente línea de código	
1.upto (9) {Planeta.new}	
Intenta entender el código proporcionado y explica el resultado obtenido	
Traduce el código anterior a Java	

Ejercicio 15. Visibilidad de métodos en Java

Sea el siguiente código en Java:

```
public class Ejercicio15 {

    public void publico(){
        System.out.println("método público");
    }
    void paquete(){
        System.out.println("método paquete");
    }
    protected void protegido(){
        System.out.println("método protegido");
    }
    private void privado(){
        System.out.println("método privado");
    }
}
```

Prueba el siguiente código en la misma clase, en otra clase del mismo paquete y en otra clase de distinto paquete:

```
public static void main(String[] args) {
    Ejercicio15 e15 = new Ejercicio15();
    e15.publico();
    e15.protegido();
    e15.paquete();
    e15.privado();
}
```

Plantéate qué ocurre en cada una de las situaciones indicadas.

Ejercicio 16. Visibilidad de variables en Java

Sea el siguiente código en Java:

```
public class Ejercicio16 {  
    private int i=0;  
    int j= 1;  
    protected int k=2;  
    public int l=3;  
}
```

Prueba el siguiente código en la misma clase, en otra clase del mismo paquete y en otra clase de distinto paquete:

```
public static void main(String[] args) {  
    Ejercicio16 e16 = new Ejercicio16();  
    System.out.println("privada : " + e16.i);  
    System.out.println("paquete : " + e16.j);  
    System.out.println("protected : " + e16.k);  
    System.out.println("public : " + e16.l);  
}
```

Plantéate qué ocurre en cada una de las situaciones indicadas.

Ejercicio 17. Ámbito de variables y métodos en Java

Sea el siguiente código en Java:

```
public class Ejercicio17 {  
    private static int j = 0;  
    private int k = 2;  
  
    public static void muestra_jk(){  
        System.out.println("variable clase " + j);  
        System.out.println("variable instancia " + k);  
    }  
    public void muestra_kj(){  
        System.out.println("variable instancia " + k);  
        System.out.println("variable clase " + j);  
    }  
    public static void main(String[] args){  
        Ejercicio17.muestra_jk();  
        Ejercicio17.j = 10;  
        Ejercicio17.muestra_jk();  
        Ejercicio17.muestra_kj();  
  
        Ejercicio17 e17 = new Ejercicio17();  
        e17.muestra_kj();  
        e17.muestra_jk();  
    }  
}
```

Trata de identificar los errores de compilación que hay y elimínalos

Ejecútalo y trata de entender qué ocurre