

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 get y el set 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: <code>private</code> , <code>package</code> y <code>public</code>	
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";
    }
    float setColor(String unColor) {
        color = unColor;
    }
    float area() {
        return ladoMenor*ladoMayor;
    }
}

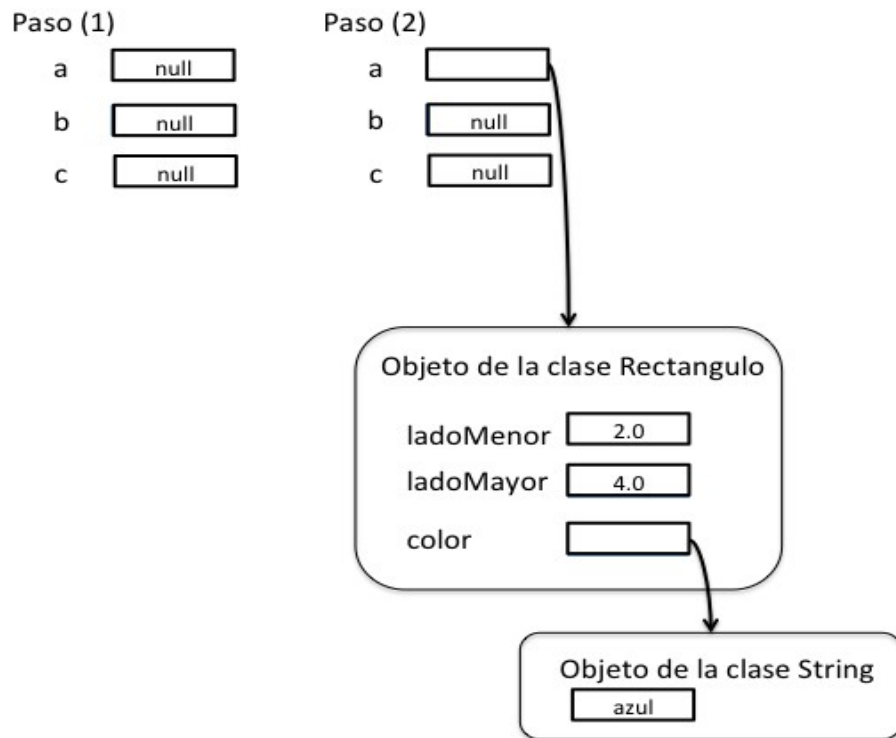
```

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;

```



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:

- a) Los atributos de clase son accesibles sólo desde métodos de clase (no desde métodos de instancia)
- b) Los atributos de instancia son accesibles sólo desde métodos de instancia (no desde métodos de clase).
- c) 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?

Ejercicio 7.

Sea la clase C.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();
```

a) ¿Qué valores tienen valor1, valor2, valor3 y valor4?

b) Traduce el código Java proporcionado a Ruby

Ejercicio 8.

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

A.java

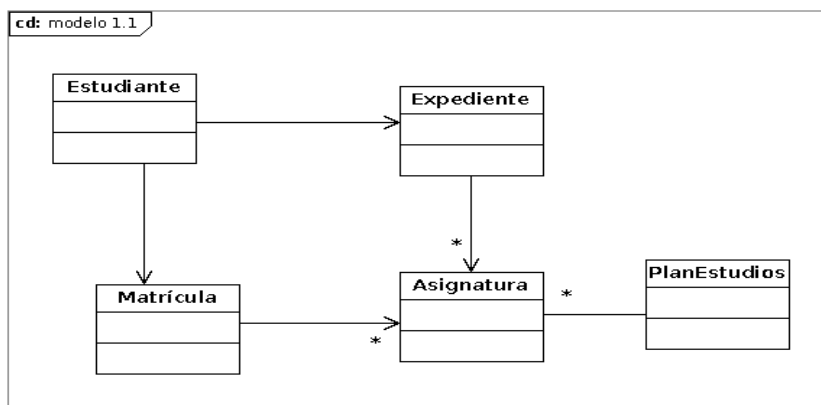
```
package p1;
import p2.B;
class A { (...) }
```

B.java

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

Responde Verdadero (V) o Falso (F)	
La clase A pertenece al paquete p1	
Como se ha hecho un import de la clase B, atributo1 es accesible desde A	
Como se ha hecho un import de la clase B, atributo2 es accesible desde A	
Traduce a Ruby el código anterior	

Ejercicio 9. A partir del siguiente diagrama de clases, obtén un esquema en el que se muestren los objetos y sus enlaces (relaciones) para el siguiente caso: 2 objetos PlanEstudios, 6 objetos Asignatura, 5 objetos Estudiante. Haz las suposiciones que consideres oportunas.



Ejercicio 10. 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];		

<code>int[] x=new int[10];</code>		
<code>Array.newInstance(int, 5);</code>		
<code>int [] dims={2,4};</code>		
<code>Array.newInstance(Alumno, dims);</code>		

Ejercicio 11. 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