



**Semana 9: Objetos**  
**Programación**  
**2018 - 2019**  
**Grado en Ingeniería Informática**

- Ejercicio 1.** Crear una clase `Telefono` con los siguientes campos: `marca`, `modelo` y `año`. En otra clase crear un objeto de este tipo y pedir al usuario que rellene los campos. Imprimir el resultado por pantalla.
- Ejercicio 2.** Crear una clase `JugadorFutbol` con los atributos `nombre`, `apellido`, `edad`, `posición` y `equipo`. Añadir un constructor que rellene dichos atributos a partir de parámetros correspondientes, comprobando que los atributos tengan el valor adecuado. En una clase aparte definir 3 objetos de esta clase de la siguiente forma:
- El primero lo creamos utilizando el constructor programado, pidiendo al usuario los valores de cada atributo.
  - El segundo lo creamos con un constructor por defecto y copiamos los valores del primero.
  - El tercero, no lo creamos, simplemente le asignamos la referencia al primer objeto y luego cambiamos sus atributos `edad` y `equipo`.
- A final se debe imprimir el contenido de los tres objetos para ver los valores finales de cada uno.
- Ejercicio 3.** Crear una clase `Estudiante` que represente a un estudiante de primer año. Sus atributos son: `nombre`, `apellido`, `nota programación`, `nota álgebra`, `nota cálculo`, `nota física`, `nota escritura`, `nota información` (elige el tipo más adecuado para cada uno de los atributos). Crea un constructor que reciba valores para cada uno de los atributos y compruebe que los rangos de los valores de las notas introducidas es el adecuado (si son incorrectos la nota se inicializa con un 0). Escribir un programa que cree un objeto de este tipo, solicite al usuario los valores de los campos por teclado e imprima por pantalla la información del estudiante.
- Ejercicio 4.** Crear un nuevo tipo de datos denominado `TrianguloRectangulo` para representar un triángulo rectángulo, tendrá dos atributos de tipo `float`: `base` y `altura`. Crear un constructor para recibir los valores de los campos que compruebe que son correctos. Crear también un programa que pida la base y la altura de un triángulo rectángulo y que guarde los datos en un objeto triángulo rectángulo. A continuación, pedir al usuario que pulse 0 para calcular el área del triángulo o 1 para calcular el perímetro. Imprimir por pantalla el resultado de calcular la elección del usuario.
- Ejercicio 5.** Crear una clase `Fecha` con los campos de `día (int)`, `mes (String)` y `año (int)`. Añadir un constructor que cree los objetos `Fecha` a partir de parámetros correspondientes. Debe comprobar que tanto `año`, como `mes` como `día` son correctos. Si no, creará un valor aleatorio para el dato incorrecto. Crear una clase `Asignatura` que contenga los campos: `nombre`, `cuatrimestre` y un array de objetos `Fecha`. En otra clase, en un método `main`, crear un objeto `Asignatura` para representar la asignatura de Programación, incluyendo los días de clase del mes de Noviembre. Imprimir el contenido de la variable como sigue:

Asignatura: Programación  
Cuatrimestre: Primero  
Días con clase:  
XX de Noviembre de 2018  
XX de Noviembre de 2018

**Ejercicio 6.** Crear una clase `Dado`, con campos `nombre` y `tiradas`, para simular una partida de juego de dados. Crear un constructor que recibe el nombre del jugador y un entero `n` que representa el número de tiradas (debe ser mayor que cero). Asignará el nombre al campo correspondiente y creará un array de `n` posiciones relleno con números entre el 1 y el 6. En otro programa se simulará una partida de dados, preguntando a cada jugador su nombre y el número de tiradas del juego. El ganador será el que tenga el mayor número de dados iguales. En caso de empate ganará el jugador cuyos dados sumen más.

**Ejercicio 7.** Realizar un programa que utilice un array de objetos para gestionar los datos de stock de una tienda de comestibles, la información a recoger será: `nombre` del producto, `precio` y `cantidad` en stock. La tienda dispone de 10 productos distintos. El programa debe ser capaz de:

1. Dar de alta un producto nuevo.
2. Buscar un producto por su nombre (y que imprima su precio y stock)
3. Modificar el stock y precio de un producto dado.

Ejemplo de ejecución:

```

Seleccione opción
1. Añadir producto
2. Buscar producto
3. Modificar stock y precio
4. Salir
1
Introduzca nombre, precio y stock
naranjas
1,2
10
Seleccione opción
1. Añadir producto
2. Buscar producto
3. Modificar stock y precio
4. Salir
1
Introduzca nombre, precio y stock
manzanas
1,5
20
Seleccione opción
1. Añadir producto
2. Buscar producto
3. Modificar stock y precio
4. Salir

```

```

2
Introduzca el nombre
manzanas
El precio es 1.5
Hay 20 unidades
Seleccione opción
1. Añadir producto
2. Buscar producto
3. Modificar stock y precio
4. Salir
3
Introduzca el nombre
manzanas
Introduzca el nuevo precio
1,6
Introduzca la nueva cantidad
19
Seleccione opción
1. Añadir producto
2. Buscar producto
3. Modificar stock y precio
4. Salir
4
Gracias por usar el programa

```

## Normas de entrega

Los ejercicios se deben **subir a Aula Global** hasta las 21:00 del Jueves 08 de Noviembre de 2018. Se deberá subir un fichero comprimido **zip** con los archivos `.java` de la carpeta **src** (se deben subir únicamente esos archivos, no el proyecto Eclipse completo). El nombre del fichero será "s9-iniciales-del-alumno1-iniciales-del-alumno2.zip" (por ejemplo Lucía Pérez Gómez y Juan García Jiménez subirán un archivo llamado s9-lpg-jgj.zip). **Ambos miembros de la pareja deberán subir el fichero.** Se debe usar el enlace de entrega del grupo de laboratorio al que se pertenezca.