

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo Título de <b>Técnico Superior en Desarrollo de Aplicaciones Web</b>			
Centro Educativo <b>IES Campanillas (sede CITIC)</b>		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>		Profesor <b>Luis José Sánchez González</b>	
Curso lectivo <b>2015 / 2016</b>	Grupo <b>1º DAW</b>	Tipo de documento <b>Examen</b>	Trimestre <b>Tercero</b>	Fecha <b>3 de mayo de 2016</b>	Modelo <b>B</b>

### INSTRUCCIONES

- ➔ El alumno debe entregar una carpeta con las soluciones al examen cuyo nombre debe estar formado por "Ex" seguido del número de lista, seguido de las iniciales. Por ejemplo, Facundo Romuedo Piladro que es el número 8 de la lista entregaría una carpeta con nombre **Ex08frp**.
- ➔ Los ficheros o carpetas correspondientes a las soluciones se deben nombrar igual que la carpeta junto con el número del ejercicio, por ejemplo **Ex08frp1.java, Ex08frp2.java, etc.**
- ➔ En los comentarios de cada programa **se debe indicar el nombre completo**, la fecha y - si procede - el modelo de examen.
- ➔ Únicamente se necesita entregar el código fuente en java, **no se deben entregar los archivos con la extensión .class**.
- ➔ Para los programas hechos en **Netbeans**, no hay que entregar la carpeta nbproject, WEB-INF, etc.

### EJERCICIOS

1. Crea un **ArrayList** de objetos de la clase **Moneda**. Las monedas se generan al azar. El número de monedas generadas es un número aleatorio entre 1 y 8 (se pueden repetir). A continuación se debe mostrar la suma total en euros y en céntimos. Si la cantidad total es menor de un euro, no se muestran los euros en la suma. Si la cantidad en céntimos es 0, tampoco se muestra.

Ejemplo 1: 2 céntimos 50 céntimos 2 euros 1 euro Suma: 3 euros y 52 céntimos	Ejemplo 2: 20 céntimos 50 céntimos 2 euros 20 céntimos 10 céntimos Suma: 3 euros	Ejemplo 3: 2 céntimos 1 céntimo 50 céntimos Suma: 53 céntimos
---	--	---

Las monedas de curso legal son las siguientes:



2. Modifica la aplicación **Gestisimal** (Gestión Simplificada de Almacén) realizada en clase mediante un **ArrayList** de tal forma que cumpla los siguientes requisitos:
  - a) Debe haber, al menos, 5 artículos precargados.
  - b) Cada artículo debe tener un atributo que indique el envasado: "brick", "enlatado", "embotellado", "a granel", etc. Al dar de alta un producto, el usuario introduce el envasado que procede (no hay unos tipos predeterminados).
  - c) Se podrá realizar un listado completo o un listado filtrado por el envasado del producto (se pide por teclado o se elige de un menú generado al efecto).

3. Implementa la clase **Empleado**. Sobre cada empleado se quiere saber su nombre, apellido y cargo que ostenta en la empresa. En principio, cuando se da de alta un empleado sólo se indican nombre y apellido. Posteriormente, se puede asignar un cargo a un empleado. La ordenación de los empleados se realiza en función del apellido; si coincide el apellido se ordena por nombre. Es necesario saber el número total de empleados dados de alta y el número de empleados que tienen un cargo asignado. El siguiente programa (se adjunta con el enunciado del examen) genera la salida que se muestra:

```
import java.util.ArrayList;
import java.util.Collections;

public class Ejercicio3 {

    public static void main(String[] args) {
        ArrayList<Empleado> e = new ArrayList<>();

        Empleado e1 = new Empleado("Elba", "Lazo");
        Empleado e2 = new Empleado("Elba", "Quero");
        Empleado e3 = new Empleado("Albert", "Quero");
        Empleado e4 = new Empleado("Mario", "Neta");
        Empleado e5 = new Empleado("Yola", "Prieto");
        Empleado e6 = new Empleado("Monica", "Galindo");
        e2.asignaCargo("Diseñador gráfico");
        e5.asignaCargo("Programador senior");
        e6.asignaCargo("Programador junior");
        e.add(e1);
        e.add(e2);
        e.add(e3);
        e.add(e4);
        e.add(e5);
        e.add(e6);

        Collections.sort(e);

        for (Empleado empleadoAux : e) {
            System.out.println(empleadoAux);
        }

        System.out.println("Empleados totales: " + Empleado.getEmpleadosTotales());
        System.out.println("Empleados con cargo: " + Empleado.getEmpleadosConCargo());
    }
}
```

```
Galindo, Monica - Programador junior
Lazo, Elba - (sin cargo)
Neta, Mario - (sin cargo)
Prieto, Yola - Programador senior
Quero, Albert - (sin cargo)
Quero, Elba - Diseñador gráfico
Empleados totales: 6
Empleados con cargo: 3
```

4. Nos han encargado realizar una aplicación sobre remedios naturales y empezaremos guardando información sobre algunas plantas de tal forma que podamos saber su nombre común, su nombre científico y sus propiedades. Crea un programa que pida al usuario el nombre común de una planta y que, a continuación, nos muestre el nombre científico y sus propiedades curativas. Si la planta introducida no existe, se mostrará el mensaje **“Lo siento, esa planta no existe”**.





Ejemplo:

Por favor, introduzca el nombre de una planta: romero

Nombre científico: rosmarinus officinalis

Propiedades: Tiene efectos estimulantes y tónicos, favorece en la recuperación de las enfermedades respiratorias y del aparato digestivo.

La información se debe guardar en dos diccionarios (dos HashMap). Uno de ellos tendrá parejas clave-valor del tipo **(nombre común, nombre científico)** y otro **(nombre común, propiedades)**. El programa no se dará por válido si no utiliza los dos **HashMap** indicados, aunque funcione bien. A continuación se muestra la información en una tabla:

	Nombre común	Nombre científico	Propiedades
	ortiga	urtica dioica	Es diurética y depurativa. Baja los niveles de colesterol.
	romero	rosmarinus officinalis	Tiene efectos estimulantes y tónicos, favorece en la recuperación de las enfermedades respiratorias y del aparato digestivo.
	aloe vera	aloe barbadensis miller	Regenera las células de la piel e incluso las de tejidos internos. Elimina hongos y virus. Protege el sistema inmunitario.
	ayahuasca	banisteriopsis caapi	Se usa en el tratamiento de la depresión y la ansiedad así como en los problemas de personalidad y esquizofrenia.