

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

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 turno.
- ➔ 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

- En HTML / CSS, un color se puede indicar mediante la palabra en inglés o su código RGB. Crea un programa que nos ayude en nuestro diseño de páginas web. El programa deberá pedir un color en español y nos dará el correspondiente color en inglés y el código RGB. Si el color introducido no existe, se mostrará el mensaje **"Lo siento, ese color no existe"**.

Ejemplo:

Por favor, introduzca un color: verde

El color verde, en inglés es Green y su código RGB es #008000.

La información se debe guardar en dos **HashMap**. Uno de ellos tendrá parejas clave-valor del tipo **(color español, color inglés)** y otro **(color español, código RGB)**. A continuación se muestra la tabla de colores:

Blue	#0000FF	azul
Black	#000000	negro
Red	#FF0000	rojo
White	#FFFFFF	blanco
Green	#008000	verde
Purple	#800080	morado
Yellow	#FFFF00	amarillo
Orange	#FFA500	naranja
Violet	#EE82EE	rosa

El programa no se dará por válido si no utiliza los dos **HashMap** indicados, aunque funcione bien.

2. Implementa la clase **Alumno**. Sobre cada alumno se quiere saber su nombre, apellido y curso en el que está matriculado. En principio, cuando se da de alta un alumno sólo se indican nombre y apellido. Posteriormente, se puede matricular a un alumno indicando el grupo. La ordenación de los alumnos se realiza en función del apellido; si coincide el apellido se ordena por nombre. Es necesario saber el número total de alumnos dados de alta y el número de alumnos matriculados. 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 Ejercicio2 {

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

        Alumno a1 = new Alumno("Elba", "Lazo");
        Alumno a2 = new Alumno("Elba", "Quero");
        a2.matricula("2º BACH");
        Alumno a3 = new Alumno("Albert", "Quero");
        Alumno a4 = new Alumno("Mario", "Neta");
        Alumno a5 = new Alumno("Yola", "Prieto");
        a5.matricula("1º DAW");
        Alumno a6 = new Alumno("Monica", "Galindo");
        a6.matricula("2º DAW");

        a.add(a1);
        a.add(a2);
        a.add(a3);
        a.add(a4);
        a.add(a5);
        a.add(a6);

        Collections.sort(a);

        for (Alumno miAlumno: a) {
            System.out.println(miAlumno);
        }

        System.out.println("Alumnos totales: " + Alumno.getAlumnosTotales());
        System.out.println("Alumnos matriculados: " + Alumno.getAlumnosMatriculados());
    }
}
```

```
Galindo, Monica - 2º DAW (matriculado)
Lazo, Elba - (no matriculado)
Neta, Mario - (no matriculado)
Prieto, Yola - 1º DAW (matriculado)
Quero, Albert - (no matriculado)
Quero, Elba - 2º BACH (matriculado)
Alumnos totales: 6
Alumnos matriculados: 3
```

3. Crea dos mazos de cartas de la baraja española generadas de forma aleatoria. Cada mazo debe ser un **ArrayList** de objetos de la clase **Carta**. A continuación, junta esos mazos en uno solo (otro ArrayList) de forma que las cartas se mezclen aleatoriamente. Tanto los dos mazos originales como la mezcla se deben mostrar por pantalla. El número de cartas de los mazos originales es un número aleatorio entre 1 y 5.

Ejemplo:

Mazo 1:

tres de bastos
caballo de copas
siete de oros

Mazo 2:

as de oros
cinco de espadas

Mezcla:

as de oros
tres de bastos
caballo de copas
cinco de espadas
siete de oros

4. 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 tipo de producto: "alimentación", "limpieza", "perfumería", etc. Al dar de alta un producto, el usuario puede introducir el tipo que quiera (no hay unos tipos predeterminados).
 - c) Se podrá realizar un listado completo o un listado filtrado por el tipo de producto (se pide por teclado o se elige de un menú generado al efecto).