



Examen DAM 2021/2022. Módulo MP0485 Programación. UD5+UD6. Tiempo estimado: 3 sesiones.

Apellidos: \_\_\_\_\_ Nombre: \_\_\_\_\_ DNI: \_\_\_\_\_

El examen se entregará en el aula virtual. Entregaremos un proyecto NetBeans, con 3 paquetes (uno por ejercicio)

El único material que se permite, será el disponible en el aula virtual. El monitor será grabado para demostración del improbable incumplimiento de esta norma.

La cualificación máxima del examen es 10.

## 1. Serialización de Objetos (4 puntos)

a) La clase **Usuarios** permite guardar en un fichero los datos de la cuenta de usuarios y leerlos. Se pide crear la clase Usuarios de modo que implemente:

- Método `main`
  - Creará 1 cuenta de usuario
  - Llamará al método “guardarCuentaFichero” que lo almacena en el fichero “usuarios.dat”
- `public static boolean guardarCuentaFichero(Cuenta cuentaUsu, String nFichero)`
- `public static Cuenta recuperarCuentaFichero(String nFichero)`

b) Además, desarrollar la clase **Cuenta** con los atributos email y fechaCreación (tipo Date), de modo que fechaCreación no se almacene en el fichero.

Añadir los métodos necesarios para que se produzca la siguiente salida al recuperar el usuario del fichero:

```
Cuenta usuario: catalina@gmail.com  
Fecha creación: no disponible
```

## 2. Eliminar objetos de ArrayList (3 puntos)

Dada una clase Artículo:

```
public class Artículo {  
  
    private String      codigo;  
    private String      descripcion;  
    private int          unidades;  
  
    public Artículo(String cod, String desc, int uni) {  
        this.codigo = cod;  
        this.descripcion = desc;  
        this.unidades = uni;  
    }  
  
    public String getCodigo() {  
        return codigo;  
    }  
}
```

Y una colección de artículos como por ejemplo:

```
ArrayList<Artículo> coleccion = new ArrayList();  
coleccion.add(new Artículo("11111A", "Pantalón", 11));  
coleccion.add(new Artículo("22222B", "Camisa", 22));  
coleccion.add(new Artículo("33333C", "Patata", 33));  
coleccion.add(new Artículo("22222B", "Camisa", 55));
```

Indica cómo implementarías el método **eliminar artículos por código** de la lista.

```
public boolean eliminarArtículo(ArrayList<Artículo> coleccion, String código)
```



### 3. Analizador léxico (3 puntos)

Supongamos que tenemos un compilador que necesita verificar si una determinada sentencia o instrucción está equilibrada en cuanto a número de paréntesis.

Para ello utilizaremos la “Collection” adecuada de Java, de modo que se emparejen paréntesis de apertura ‘(’ con paréntesis de cierre ‘)’.

Tendremos que ir analizando todos los caracteres de la expresión y actuar cuando detectemos un paréntesis:

- añadiéndolo a nuestra colección si es de apertura,
- y retirándolo si es de cierre.

Para ello crearemos la clase AnalizadorLexico:

```
public static void main(String arg[]) {  
  
    String cadena_no1 = "(Cadena no equilibrada en paréntesis((()()())))";  
    String cadena_no2 = ")Cadena no equilibrada en paréntesis(";  
  
    String cadena_si1 = "(Cadena equilibrada en parentesis())";  
    String cadena_s2 = "(Cadena equilibrada ((en)) parentesis())";  
    // Llamada a verificaParentesis de las 4 cadenas  
}  
  
private static boolean verificaParentesis(String cadena) {  
  
    //Resolver aquí el método que compruebe que la cadena está equilibrada  
}
```

*La mejor forma de predecir el futuro es crearlo.*

Peter Drucker