

Objetivos:

- Repaso al lenguaje de programación Java (Transparencias)
- Desarrollo implementación de varios ejemplos java, clases en Java y Array de objetos Java

Contenido:

Entorno de desarrollo

Para el desarrollo de las prácticas se utilizará el entorno Eclipse.

Ejemplo 1. Clase Persona

Notas: En la implementación de cualquier clase deben implementarse los siguientes métodos:

- Constructores: por defecto, copia y parametrizado/s
- Selectores: getX() donde X es todo atributo establecido en la clase
- Modificadores: setX() donde X es todo atributo establecido en la clase
- equals(): para comparar objetos de la clase
- toString(): Devolver la información asociada a un objeto

Importante: En todas las entregas de este año es obligatorio utilizar Javadoc en el código fuente así documentar internamente el código

```
/**
 * Implementacion de la clase Persona formada por el nombre, el dni y la edad.
 *
 * @author Luis Arévalo
 */
```

```
package es.unex.mdp.practica0;
```

```
public class Persona {
```

```
    private String nombre;
    private String dni;
    private int edad;
```

```
    /**
     * Constructor por defecto
     */
```

```
    public Persona() {
        nombre = null;
        dni = null;
        edad = 0;
    }
```

```
    /**
     * Constructor parametrizado
     *
     * @param nombre Nombre de la persona
```

```

* @param dni DNI de la persona
* @param edad Edad de la persona
*/
public Persona(String nombre, String dni, int edad) {
    this.nombre = nombre;
    this.dni = dni;
    this.edad = edad;
}

/**
 * Constructor de copia
 *
 * @param p Persona
 */
public Persona(Persona p) {
    nombre = p.nombre;
    dni = p.dni;
    edad = p.edad;
}

/**
 * Devuelve el nombre de la persona
 * @return El nombre de la persona
 */
protected String getNombre() {
    return nombre;
}

/**
 * Establece el nombre de la persona
 * @param nombre El nombre de la persona
 */
protected void setNombre(String nombre) {
    this.nombre = nombre;
}

/**
 * Devuelve el dni de la persona
 * @return DNI de la persona
 */
protected String getDni() {
    return dni;
}

/**
 * Establece el dni de la persona
 * @param dni DNI de la persona
 */
protected void setDni(String dni) {
    this.dni = dni;
}

/**
 * Devuelve la edad de la persona
 * @return Edad de la persona
 */
protected int getEdad() {
    return edad;
}

```

```

/**
 * Establece la edad de la persona
 * @param edad Edad de la persona
 */
protected void setEdad(int edad) {
    this.edad = edad;
}

/**
 * Recupera en formato String los valores de los atributos de la clase
 * toString
 */
@Override
public String toString() {
    return "Persona [nombre=" + nombre + ", dni=" + dni + ", edad=" + edad
        + "];"
}

/**
 * Compara dos personas por DNI
 */
@Override
public boolean equals(Object o) {
    Persona p = (Persona) o;
    /* MAL, pues dni es un String y estaríamos comparando sus referencias
    * if (dni==p.dni) return true;
    * else return false;
    */
    /* BIEN */
    return dni.equals(p.dni);
}
}

```

Ejemplo 2. Ejemplo de uso de la Clase Persona

```

/**
 * Programa principal para usar la clase Persona
 */
package es.unex.mdp.practica0;

/**
 * @author Luis Arévalo
 */
public class Main {

    /**
     * @param args Parametros de entrada de la consola
     */
    public static void main(String[] args) {

        Persona p = new Persona();
        Persona p1 = new Persona("Paco", "12345678A", 37);
        Persona p2 = new Persona(p1);
        System.out.println(p);
        System.out.println(p1);

        Persona p3 = p2;
        p3.setEdad(100);
        System.out.println(p2);
        System.out.println(p3);
    }
}

```

```

        Persona p4 = new Persona(p2.getNombre(), p2.getDni(), p2.getEdad());

        Persona[] vector = new Persona[10];

        vector[0] = new Persona();
        vector[1] = new Persona("", "", 10);
        vector[2] = p4;
    }
}

```

Ejercicio a entregar

¿Cómo se realizará la entrega?

Las entregas se realizarán en el campusvirtual alternativo (<http://vpl-cum.unex.es/moodle>) en la asignatura de Metodología y Desarrollo de Programas. Se trata de un espacio virtual donde existe una tarea que evalúa automáticamente las entregas proporcionando una calificación inmediata. Este plugin del campusvirtual alternativo tiene las siguientes particularidades:

- La evaluación se realiza en base a las entradas por teclado y a las salidas esperadas, es decir, si se esta implementando una calculadora y se recibe “2 4 +”, a partir de los valores de entrada el resultado esperado sería 6, de tal forma que cualquier otro valor implicaría una implementación incorrecta de la calculadora.
 - Cuidado: Si la salida es “Suma: 10” las siguientes salidas serán erróneas: “suma: 10” (minúscula), “Suma:10”(sin espacio), “Suma: 10 “ (espacio al final), etc.
- NO SE puede realizar ningún syso en la introducción de valores por teclado, los únicos syso permitidos son los que se espera en la salida. Es decir, si se desea pedir un valor String sería:

```

Scanner in = new Scanner ( System.in );
syso("Dame la edad");
String n= in.nextLine();

```

- Se recomienda antes de subir la entrega, probarla en un ordenador local. Por el punto anterior, la ejecución puede ser muy tediosa, por lo que se recomienda utilizar la clase **Teclado** que se proporciona:


```
public class Teclado {
    ....
    private boolean CV= false; //Este atributo estará a false cuando se hagan las pruebas locales (en
                                //ordenador) y se pondrá a true cuando se suba al campusvirtual
    ..
}
```
- La lectura por teclado se debe realizar si tenemos un objeto Teclado t de la siguiente forma:
 - Para leer cadena: String x= t.literalConString(“Dame el nombre”);
 - Para leer enteros: int x=t.literalConEntero(“Dame la edad”);
 -
- Hay que tener cuidado con la codificación de los caracteres. En principio esta configurado para que soporte acentos (sólo en los comentarios) y ñ (aunque la ñ NUNCA se debe usar en programación). Si se observa que en una ejecución da fallo por un acento, quitarlos y avisar al profesorado.
- Cuidado con el siguiente ejemplo, **sólo puede haber una instancia Teclado**

Código que falla en CV alternativo, no en Eclipse

```
public class XXXX(){
    public void yyyy(){
        Teclado t= new Teclado();
        ....
    }
    public void zzzz(){
        Teclado t= new Teclado();
        ....
    }
}
```

Solución

```
public class XXXX(){
    private Teclado t= new Teclado();
    public void yyyy(){
        ....
    }
    public void zzzz(){
        ....
    }
}
```

¿Qué se debe entregar? (Mirar siempre el enunciado del CV alternativo)

Cada alumno deberá implementar una clase que se le ocurra distinta que este formada:

- Un atributo String, un atributo entero y un atributo float (en este orden)
- Los constructores: por defecto y al menos un parametrizado
- Los accesores y mutadores
- El método toString() y equals()

IMPORTANTE: El código fuente tiene que estar en un paquete denominado: **es.unex.cum.mdp.sesion0**

A continuación se realizará una clase principal denominada **Main.java** donde se definirá un vector de esa clase e implementar las siguientes operaciones:

1. Inicializar el vector: Se reserva tamaño para el array de objetos pidiendo para ello por teclado el tamaño del array.
2. Rellenar el vector con tanto objetos como tamaño del vector. Se pedirá por teclado en el orden indicado (String, entero, float) y cuando se suba sin realizar ningún syso
3. Listar todos los objetos del vector en el siguiente formato por cada objeto
[String, int, float] es decir, podría ser algo como: [renault, 110, 18000.45]
4. Obtener la suma y la media del campo de tipo entero con el siguiente formato:
Suma: XXX Media: YYY

Donde XXX es la suma de los valores numéricos de los objetos del vector e YYY es la media.

5. Leer un valor float y comprobar si se encuentra en el array. La salida será la siguiente siendo XX el valor introducido:

Si esta → “El valor XX se encuentra en el array”

Si no esta → “El valor XX no se encuentra en el array”

6. Mostrar el valor más alto del array del campo entero mostrando “El valor mas alto es XXX y esta en la posicion YYY” donde XXX es el valor e YYY es su posición dentro del array. Si hay dos valores o más valores iguales, se seleccionará el último de ellos.