


Prueba Teórico-Práctica 2º Trimestre - Tiempo 3h	9 de Marzo de 2020
Nombre y Apellidos	DNI/NIE: Firma:
1º Desarrollo de Aplicaciones Web (Vespertino) Módulo: Programación	 IES Alonso de Avellaneda (Alcalá)

Programas (10 puntos) (Recogida de los fuentes del programa en repositorio github de cada alumno)

Ejercicio 1 (2.5 puntos)

Utilizando arrays, escribe un programa que lea números enteros por teclado del 1 al 100 y cuente las ocurrencias de cada uno de ellos. La entrada de usuario se para cuando encuentra un 0.

```

Introduzca enteros entre 1 y 100: 2 5 6 5 4 3 23 43 2 0
2 sale 2 veces
3 sale 1 vez
4 sale 1 vez
5 sale 2 veces
6 sale 1 vez
23 sale 1 vez
43 sale 1 vez

```

Ejercicio 2 (2.5 puntos)

En una empresa queremos saber qué horas totales por empleado se han trabajado. Para ello se almacenan las horas de los empleados con la tabla de más abajo. Utilizando arrays unidimensionales y/o bidimensionales para crear la salida con dos columnas empleados y suma total de horas, realizar el programa que implemente lo especificado.

	Do	Lu	Ma	Mi	Ju	Vi	Sá
Empleado 0	2	4	3	4	5	8	8
Empleado 1	7	3	4	3	3	4	4
Empleado 2	3	3	4	3	3	2	2
Empleado 3	9	3	4	7	3	4	1
Empleado 4	3	5	4	3	6	3	8
Empleado 5	3	4	4	6	3	4	4
Empleado 6	3	7	4	8	3	8	4
Empleado 7	6	3	5	9	2	7	9

Output:

Empleado	Horas semanales
----------	-----------------

0	34
---	----

1	28
---	----

2	20
---	----

3	31
---	----

4	32
---	----

5	28
---	----

6	37
---	----

7	41
---	----

Las cabeceras de los métodos utilizados:

```
public static int[][] sumFilas(int[][] m) -> suma las filas
public static void print(int[][] m) -> imprime matriz
```

Ejercicio 3 (2.5 puntos)

Implementa un programa que imprima y calcule la suma de la diagonal secundaria de una matriz $n \times n$. Las matrices deben ser introducidas por teclado con $n \geq 3$ con números aleatorios entre [0-9]. Se adjuntan las siguientes cabeceras de los métodos:

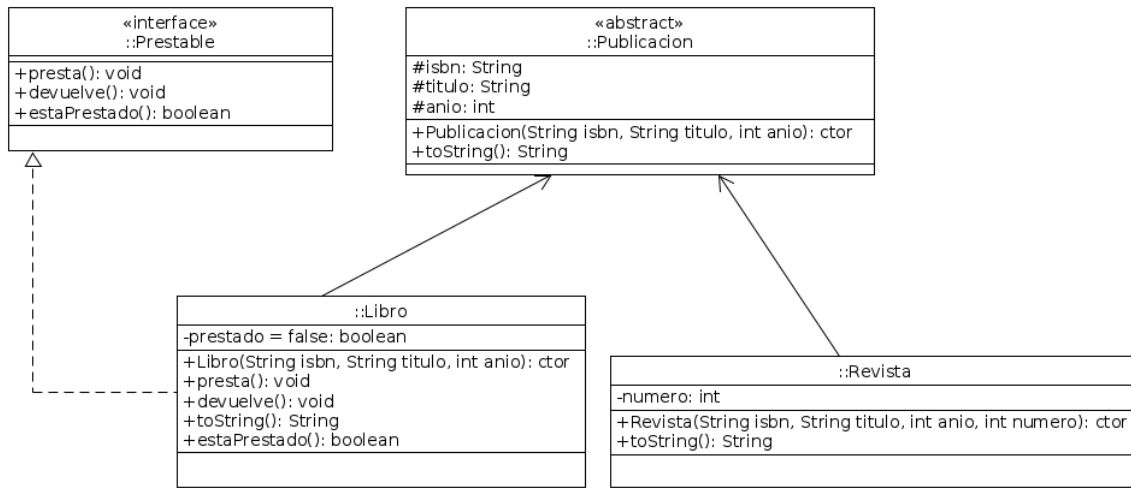
```
public static double[][] setMatrix(int n)
public static void getMatrix(double[][] m)
public static double getMedia2DSec(double[][] m)
```

Output:

```
Introduzca la dimensión n x n: 3
8.0 2.0 2.0
5.0 9.0 5.0
0.0 4.0 9.0
Cálculo media secundaria:
2.0 9.0 0.0 Media es 3,67:
```

Ejercicio 4 (2.5 puntos)

Se tiene una biblioteca donde existen libros y revistas los cuales son publicaciones y tienen como atributos comunes el isbn, título y año de publicación. No obstante, los libros tienen la característica de que son prestables, es decir, que se puede prestar, devolver y consultar su estado (prestable o no). Mientras que las revistas no lo son. Aquí se presenta un diagrama UML con el diseño.



Publicación: establece los atributos y declara método `toString()` como abstract.

Prestable: establece el interfaz con métodos

Libro: establece los atributos e implementa los métodos convenientemente. El método `toString()` debe devolver el isbn, título, año y si está prestado o no.

Revista: Inicializa todos sus atributos y `toString()` debe devolver: isbn, título, año y número

Se tiene que realizar una clase **TestBiblioteca** que realice lo siguiente:

- Cree un array de 3 objetos Libro e inicialice sus atributos
- Cree un array de 2 objetos Revista e inicialice sus atributos
- Imprima ambos arrays
- Presta libro ISBN:456789 y consulta estado. Intenta prestarlo.
- Presta libro ISBN:112233 y devuelve. Consulta estado

Output:

```

ISBN: 123456, título: Programación en Java, año de publicación: 2018; no
prestado
ISBN: 112233, título: Metodología de la Programación, año de publicación: 2016;
no prestado
ISBN: 456789, título: Sistemas Operativos, año de publicación: 2015; no prestado
Revista con isbn 444555título: Linux Magazineañ número: 344
Revista con isbn 002244título: Cómo Programaraño número: 255
El libro está prestado
Lo siento, ese libro ya está prestado.
  
```

Rúbrica

Todos los programas, métodos, etc. deben estar bien especificados. Las salidas no puntúan por separado sino que son una consecuencia del funcionamiento de métodos, programas, etc.

```

E1: 2p especificación. Salida 0.5p
E2: 1.5p sumFilas. 0.5 print. 0.5 Si la salida es correcta
E3: 1.5p: getMedia2DSec; 0.5: setMatrix; 0.25 : getMatrix. Salida : 0.25
E4: Jerarquía con métodos y atributos completos 2p. TestBiblioteca: 0.5
  
```