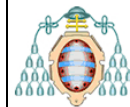




Apellidos:

Nombre:

D.N.I.:



INTRODUCCIÓN A LA PROGRAMACIÓN - E. P. DE INGENIERÍA DE GIJÓN
19 DE ENERO DE 2018

[1] (0.5 puntos) Declara una constante no estática para la Constante de Legendre (valor 1.08366)

--

[2] (1,5 p.) Dada la siguiente definición de variables y sus valores iniciales,

int x=9, y=2; double f=3.5; boolean b=true;

indica para las siguientes expresiones, si son o no correctas (SI/NO), en caso de resultar incorrectas JUSTIFICA por qué lo son, y en el caso de ser correctas indica el TIPO y el VALOR que producen:

++f	<input type="checkbox"/> Sí <input type="checkbox"/> No Tipo-Valor/Motivo:
b < x	<input type="checkbox"/> Sí <input type="checkbox"/> No T-V/M:
f = x/y	<input type="checkbox"/> Sí <input type="checkbox"/> No T-V/M:
x>y && b	<input type="checkbox"/> Sí <input type="checkbox"/> No T-V/M:
x <> y	<input type="checkbox"/> Sí <input type="checkbox"/> No T-V/M:
x+b	<input type="checkbox"/> Sí <input type="checkbox"/> No T-V/M:

[3] (2 p.) Supuesto que $N > 0$ y **par**, diseña los algoritmos iterativos que se indican a continuación.

Bucle for que imprima una secuencia de números en $[1, N]$ donde cada número sea el doble del anterior. Ej N=18: 1 2 4 8 16	Bucle do/while para imprimir la secuencia $N \ N/2 \ (N/2)/2 \dots$ hasta obtener un impar (no incluido) Ej: N=40: 40 20 10	Bucle while para imprimir la secuencia de números de $N/2$ a 1, ambos incluidos Ej: N=8: 4 3 2 1

[4] (3 p.) Escribir el método **rotarColumnaArriba**, privado y estático, que recibe una matriz de caracteres no vacía y el índice de una de sus columnas y modifica la matriz rotando las componentes de esa columna una posición hacia arriba. Se supone que el índice de la columna es correcto.

Ejemplo:

rotarColumnaArriba(M,0)

M

'O'	'O'	'O'	'X'
'X'	'X'	'X'	'O'
'O'	'O'	'X'	'O'
'X'	'X'	'O'	'X'

M

'X'	'O'	'O'	'X'
'O'	'X'	'X'	'O'
'X'	'O'	'X'	'O'
'O'	'X'	'O'	'X'

--

[5] (1.5 p.) Sustituye las sentencias siguientes por otro trozo de código equivalente (que produzca los mismos resultados) pero que sea más corto, más fácil de entender y/o más eficiente.

<pre>if (x<18) d--; else if (x>65) d--; else if (x<=65) d++;</pre>	<pre>if (y<-10) if (y>10) d=d/10; else d=d*10; else d=d*10;</pre>	<pre>if (z>=5) { x++; y*=2;} else { x++; y/=2;}</pre>

[6] (3.5 p.) El Club de Tenis de Gijón va a organizar un torneo en el que los precios de la inscripción son los que se muestran en la tabla adjunta. Se pide hacer un **programa de consola completo** (importando las clases necesarias) que pida por teclado el tipo de competición y de jugador/a y nos indique el precio de la inscripción. Se valorará especialmente el diseño del esquema condicional, y la conveniencia y el correcto uso de las variables y tipos elegidos para representar los datos. El usuario introducirá los datos correctamente, no se debe incluir código para comprobar si son correctos.

	Alevín	Infantil	Junior	Senior
Individuales	20	20	25	25
Dobles	10	10	20	20
Dobles Mixtos	10	10	15	15

[7] (2 p.) Escribir el método estático **calculaPicos** que reciba como parámetro un vector de enteros y que devuelva el número de picos que contiene. Un pico es un elemento que es estrictamente mayor que las componentes anterior y siguiente.

Ejemplo:

1	3	0	5	7	9	4	1	5
---	---	---	---	---	---	---	---	---

Retorna: 2. Hay 2 picos: 3, y 9

[8] (3 p.) Escribir el método público y estático **dobleLetra** que reciba como parámetro una palabra (representada con una cadena de caracteres) y que devuelva cierto si en la palabra al menos una misma letra aparece en dos posiciones consecutivas. Debe devolver falso en caso contrario.

Ejemplos: "lluvia" "acción" "arrollar"

Restricción: **no se pueden emplear** las sentencias break y return dentro de un bucle.

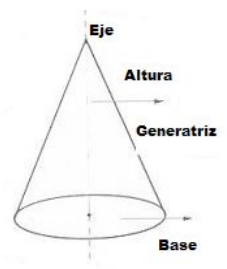
[9] (4 p.) Escribir la clase **Cono** para representar conos. Un Cono se identifica por su radio y su altura, que son valores de tipo real. Deben implementarse los siguientes métodos:

- Tres constructores: 1) sin parámetros (ambos valores valdrán 1.0), con dos parámetros reales (uno para inicializar cada valor), y el constructor copia.
- Métodos set() y get() para los atributos, con la restricción de que ambos deben tener siempre una cantidad mayor que 0.
- Métodos calculaGeneratriz(), calculaÁrea() y calculaVolumen() (ver fórmulas).

$$g = \sqrt{a^2 + r^2} \quad \text{area} = \pi r^2 + \pi r g \quad \text{volumen} = \frac{\pi r^2 a}{3}$$

- Método esMayor(), que recibe como parámetro otro objeto Cono y devuelve cierto si el objeto que invoca el método tiene un volumen mayor que el del objeto pasado como argumento.

Notas: Se tendrá en cuenta la corrección de la implementación y que la clase se escriba de forma que se faciliten futuras modificaciones. No es necesario poner comentarios Javadoc.



[10] (3 p.) Escribir la clase **Tarjeta** que representa los datos de una tarjeta de crédito y las operaciones que se pueden realizar con ella. Los datos que tiene son:

- El **nombre** del titular: cadena de caracteres.
- El **número** de la tarjeta: 16 dígitos.
- La **fecha** de validez: mes y año. Deben usarse obligatoriamente un objeto de la clase Fecha cuya especificación UML es la mostrada.
- El **límite**: cantidad en euros máxima que se puede cargar, y
- El **saldo**: cantidad gastada (valor positivo), o abonada (valor negativo).

Los métodos a implementar deben hacer funcionar el código siguiente:

```
Tarjeta t1=new Tarjeta("Juan",6023567921123217L,7, 2016 , 2500.0);
                        //nombre,          n°           ,mes, año ,límite, saldo=0
t1.cargarCompra(359.99); //saldo = 359.99
t1.abonarDevolución(39.95); //saldo = 320.04
t1.cargarCompra(2250.0); //supera el límite, no se hace!!! saldo=320.04
System.out.print(t1);   //Nombre: Juan
                        //Número: 6023567921123217
                        //Valida: 07/2016
                        //Límite: 2500.00 €      Saldo: 320.04 €
```

Notas: No hay que implementar los métodos get() y set(), **pero debes usarlos** como si lo estuviesen para acceder a los atributos en el resto de métodos. En el constructor, el campo Fecha **NO** hay que inicializarlo con su método set; si lo inicializas de ese modo debes programar dicho método. En cargarCompra() y abonarDevolución() tiene que comprobarse que la cantidad es siempre positiva.

Fecha
- mes: int - año: int
+ Fecha(int,int) + getMes(): int + setMes(int) + getAño(): int + setAño(int) + toString(): String