Apellidos:

Nombre: D.N.I.:

Introducción a la Programación - E.P. de Ingeniería de Gijón

17 de Enero de 2019

1. (1 p) Suponiendo que ya has declarado una variable de tipo entero x, declara, *con una sola instrucción*, un vector que contenga los siguientes valores reales: el mismo valor que el de x, la mitad del valor de x, el valor constante 3,1416.

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

```
String s=new String("Hola"), t=new String("Hola"), r='5'; int x=5, y=7;
```

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.

Nota: Los dígitos, al igual que las letras del alfabeto, ocupan posiciones consecutivas en la tabla de códigos.

		Tipo	Valor	Motivo
`s == t`	☐Si ☐No			
`12 = x + y`	☐Si ☐No			
`r - x == '0'`	☐Si ☐No			
`x = y = s + t`	☐Si ☐No			

3. (3 p) Suponiendo que $N \geq 0$ es **par** indica cuántos asteriscos imprime cada uno de los bucles siguientes, así como el valor de la variable i **a la salida del bucle**.

`int i = 1;`
`while (i != N) {`
`  System.out.print('*');`
`  i+=2;`
`}`
	`for (int i = 1; i <= 2*N; i+=2) {`
	`  System.out.print('*');`
	`}`
		`int i = 0;`
		`do {`
		`  i += 2;`
		`  System.out.print('*');`
		`} while (i <= N);`

¿Cuántos asteriscos? ¿Valor de i?	¿Cuántos asteriscos? ¿Valor de i?	¿Cuántos asteriscos? ¿Valor de i?

4. (6 p) Implementa un método `seno(x, n)` que calcule de forma aproximada el seno del ángulo x (expresado en radianes) utilizando los n primeros términos de la siguiente serie de McLaurin:

$$\text{sen}(x) = \sum_{i=1}^{\infty} \frac{(-1)^{i-1}}{(2i-1)!} x^{2i-1} = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \dots$$

Notas: Atención a la alternancia de signo en la suma de términos. Puedes utilizar `Math.pow()` y `factorial()`, aunque éste último deberás implementarlo también.

5. (2 p) Dadas las variables enteras i, j y k, y utilizando sintáxis de lenguaje Java escribe las condiciones booleanas equivalentes a las siguientes expresiones:

   El valor de i está en el intervalo $[j, k]$

   i es múltiplo de j pero j no es divisor de k

   i, j y k son las longitudes de los lados de un triángulo (la suma de dos cualesquiera es mayor que la tercera)

   Recorrer una distancia de j cm. requiere dar k pasos de i cm. cada uno y todavía nos quedarían 8 cm. por recorrer

6. (6 p) Una empresa de transporte aplica distintas tarifas dependiendo del tipo de carga y la zona de destino según la siguiente tabla. Se pide hacer un **programa de consola completo** (importando las clases que necesites) que pida por teclado los valores de **tipo** y **zona** e imprima la tarifa que debe aplicarse (A, B, D, E o F). Se valorará especialmente el diseño del esquema condicional.
   Nota: No hace falta incluir código para comprobar sin son correctos.

Tipo Carga \ Zona Destino	**1**	**2**	**3**	**4**	**5**
1	A	B	B	B	B
2	A	F	F	F	C
3	A	E	F	F	C
4	A	E	E	F	C
5	A	D	D	D	C

7. (6 p) Escribe el método público y estático **esHexadecimal()** que recibe un objeto String y retorna cierto si la cadena representa un número en base 16, o falso en caso contrario. Para representar un número en hexadecimal necesitamos los dígitos (0 al 9) así como las letras de la A a la F. Ejemplos: "50" retorna cierto, "J673" retorna falso, "b" retorna cierto, "A1Fc" retorna cierto. Puedes crear métodos adicionales si lo consideras apropiado.

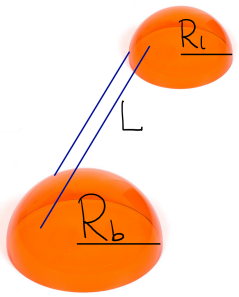
8. (6 p) Escribir el método público y estático **lt2vector()**, que recibe una matriz cuadrada de reales y retorna un vector de reales con los valores del triángulo inferior de la matriz. El vector debe tener la dimensión adecuada y se rellenará recorriendo la matriz por filas, tal como se muestra en el ejemplo:

$$M = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \quad \Longrightarrow \quad V = \begin{pmatrix} 5 & 9 & 10 & 13 & 14 & 15 \end{pmatrix}$$

9. (8 p) Implementa la clase `LámparaDespacho` para representar lámparas de escritorio con la forma que se muestra en la figura. Cada lámpara viene determinada por tres valores dados en milímetros: `Rb`$\in [50, 90]$, `Rl`$\in [25, 40]$ y `L`$\in [120, 500]$.

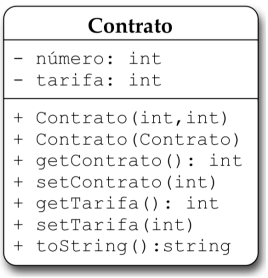
   Deben implementarse los siguientes métodos públicos:

   - Cuatro constructores: *constructor por defecto*, con los valores mínimos en cada atributo; *con un parámetro*, que será el valor para `Rb` (los demás atributos tendrán el mínimo valor posible); *con tres parámetros*; y constructor *copia*.
   - Métodos `set()` y `get()`. Cuando se suministre un valor fuera del rango permitido para cualquier atributo, éste tomará el mínimo valor posible.
   - Métodos `áreaBase()` y `áreaLuz()` que retornan la superficie de la semiesfera correspondiente (área de una esfera: $4\pi r^2$).
   - Método `precioVentaRecomendado()`, que retorna el coste de los materiales de la lámpara más un 10 % de beneficio empresarial. El material de los brazos cuesta 38€/m y el de las planchas para hacer las semiesferas cuesta 50€/m^2.
   - Método `másBarata()`, para comparar el precio de venta recomendado entre dos lámparas. Ejemplo: `l1.másBarata(l2)` retorna `true` si el precio de `l1` < precio de `l2`.



10. (6 p) Escribir la clase `Cliente` que representa los datos de un cliente de una compañia de luz y gas. Los datos que tiene un objeto `Cliente` son:
    - El `nombre` del titular: cadena de caracteres.
    - La `dirección` de su vivienda: cadena de caracteres.
    - Los atributos `luz` y `gas`: son objetos de la clase `Contrato`. Tiene dos atributos: el número de contrato y el código de la tarifa, ambos enteros.
    - El consumo: cantidad en euros gastada por el cliente.

    Notas: No hay que implementar los métodos `get()` y `set()`, pero debes usarlos para acceder a los atributos en el resto de métodos como si existiesen. En el constructor, los atributos luz y gas NO hay que inicializarlos con su método `set` correspondiente; si lo haces así, entonces debes programar ademas dichos métodos. `cargarFactura()` y `cambiarTarifa()` tienen que comprobar que el núm. de contrato es del cliente y el importe positivo. Hay que emplear siempre los métodos de la clase `Contrato`, por ejemplo, `toString()` con el atributo `luz` de `c1` imprimiría: "No: 123879 Tarifa: 2".

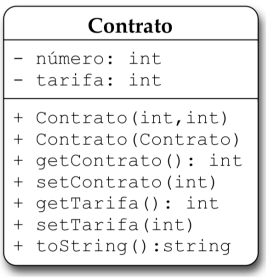


Los metodos a implementar deben hacer funcionar el codigo siguiente:

```
Cliente c1=new Cliente("Ana","Uría,6 2ºI",123879, 2, 345536,4);
                        //nombre, dirección, núm. Luz, tarifa Luz, núm. Gas, tarifa Gas
c1.cargarFactura(123879,35.6); // consumoTotal = 35.6
c1.cargarFactura(346456,124.96); //no se carga, no coincide núm de contrato
c1.cambiarTarifa(345536,3); //la tarifa de gas pasa a ser 3
System.out.print(c1); //Nombre: Ana Dirección: Uría 6, 2ºI
                      //Contrato Luz No: 123879 Tarifa: 2
                      //Contrato Gas No: 345536 Tarifa: 3
                      //Consumo: 35.6
```

10. (6 p) Escribir la clase `Cliente` que representa los datos de un cliente de una compañia de luz y gas. Los datos que tiene un objeto `Cliente` son:
    - El `nombre` del titular: cadena de caracteres.
    - La `dirección` de su vivienda: cadena de caracteres.
    - Los atributos `luz` y `gas`: son objetos de la clase `Contrato`. Tiene dos atributos: el número de contrato y el código de la tarifa, ambos enteros.
    - El consumo: cantidad en euros gastada por el cliente.

    Notas: No hay que implementar los métodos `get()` y `set()`, pero debes usarlos para acceder a los atributos en el resto de métodos como si existiesen. En el constructor, los atributos luz y gas NO hay que inicializarlos con su método `set` correspondiente; si lo haces así, entonces debes programar ademas dichos métodos. `cargarFactura()` y `cambiarTarifa()` tienen que comprobar que el núm. de contrato es del cliente y el importe positivo. Hay que emplear siempre los métodos de la clase `Contrato`, por ejemplo, `toString()` con el atributo `luz` de `c1` imprimiría: "No: 123879 Tarifa: 2".



Los metodos a implementar deben hacer funcionar el codigo siguiente:

```
Cliente c1=new Cliente("Ana","Uría,6 2ºI",123879, 2, 345536,4);
                        //nombre, dirección, núm. Luz, tarifa Luz, núm. Gas, tarifa Gas
c1.cargarFactura(123879,35.6); // consumoTotal = 35.6
c1.cargarFactura(346456,124.96); //no se carga, no coincide núm de contrato
c1.cambiarTarifa(345536,3); //la tarifa de gas pasa a ser 3
System.out.print(c1); //Nombre: Ana Dirección: Uría 6, 2ºI
                      //Contrato Luz No: 123879 Tarifa: 2
                      //Contrato Gas No: 345536 Tarifa: 3
                      //Consumo: 35.6
```