

EXAMEN DE SERVICIOS DE RED E INTERNET (JAVA)

IES LOS MANANTIALES

TORREMOLINOS 29 de noviembre de 2019

- 1) (2,5 puntos) Tenemos 2 clases de alumnos y queremos realizar un estudio sobre las notas de matemáticas han obtenido en un examen, que son las siguientes:

NOTA	1	2	3	4	5	6	7	8	9	10
ALUMNOS CLASE A	0	1	2	6	6	5	2	2	0	1
ALUMNOS CLASE B	1	3	1	3	4	2	1	1	1	1

Se pide crear un programa Java llamado ejercicio1 en el que deberás:

- Crear una tabla con dichos datos numéricos.
 - Calcular el total de alumnos de la clase A y de la Clase B, así como el total de alumnos con cada una de las distintas calificaciones.
 - Calcula la media de cada curso.
 - Calcula la media del total de datos
 - Muestra dichos datos en pantalla de la forma más ordenada posible representando la matriz en su totalidad y las respuestas a dichos enunciados.
- 2) (2,5 puntos) Un número n es abundante si la suma de todos sus divisores, incluido el propio número, es mayor que $2n$. Por ejemplo, 6 no es abundante porque los divisores de 6 son 1,2,3 y 6 ($1+2+3+6=12$ que no es mayor que $6*2=12$) sin embargo 12 sí es abundante porque sus divisores son $1+2+3+4+6+12=28 > 12*2$. Crea una función `bool abundante(int)` que dado un número entero devuelva verdadero si es abundante y falso si no lo es.
- Crea un programa llamado Abundantes que, usando la función abundante, calcule todos los números abundantes en el intervalo 1-1000, los vaya guardando en una matriz `m` y que finalmente los muestre por pantalla.
- 3) (5 puntos) Escriba un programa que implemente las clases Bombilla, Casa y UsaCasa. La clase Bombilla sirve para representar las bombillas que puede haber en una casa. Cada bombilla tiene asociada 4 atributos: sitio (String que representa el lugar de la casa donde está la bombilla) , estado (boolean) que representa encendida (True) o apagada (False), potencia (int) que indica la potencia en vatios de la bombilla y tiempo (int) que indica el tiempo en horas que ha estado la bombilla encendida. Para realizar la clase Bombilla deberá implementar:
- Un constructor `Bombilla(String,boolean)`. Por ejemplo `Bombilla("Salón",False)`, que indica que hemos creado una bombilla en el salón apagada. No se pueden crear bombillas con la cadena vacía (error). Consumo vale
 - Otro constructor `Bombilla(String,boolean,int,int)` que crea una bombilla con todos sus atributos. Ejemplo `Bombilla("Cocina",True,60,100)` crea una bombilla en la cocina que está encendida, con 60w de potencia y 100 horas encendida.
 - Métodos para cambiar todos los atributos de una bombilla (getters y setters).

- d) Cree un método llamado `coste` que devuelve un `float` para ver el consumo de cada bombilla considerando que el precio de `kwh` es de 0,20€. Por ejemplo, una bombilla que ha estado 100 horas encendida, con potencia de 60w consume

$$\text{consumo} = \text{PrecioKwh} * N^{\circ} \text{ horas} * \frac{\text{potenciaenwattios}}{1000}$$

- e) Crea un método `toString` para escribir una bombilla con el siguiente formato:
Bombilla(Lugar: Cocina, Encendida: ON, Potencia: 60w, Tiempo: 100 horas)

La clase `Casa` tiene un atributo llamado `bombillas` que es una matriz de 10 bombillas. Para realizar la clase `Bombilla` deberá implementar:

- a) Un constructor que inicialice las 10 bombillas de forma aleatoria, los posibles lugares de una casa son `String[] lugares={"Salón","Cocina","Baño","Dormitorio 1","Dormitorio 2","Dormitorio 3","Entrada","Balcón"}`.
- b) Un método llamado `consumoTotal` que calculará el consumo de TODAS las bombillas de una casa en €.
- c) Un método `toString` de `Casa` que muestre la casa de la siguiente forma:
`Casa[Bombilla 1: Bombilla(...), Bombilla 2: Bombilla(...),...,Consumo: 87,48€]`

Finalmente, crea la clase `UsaCasa` que deberá crear una matriz con 5 casas. El programa deberá llenar la matriz con 5 casas creadas aleatoriamente de la forma indicada anteriormente y mostrará en pantalla lo que hay en cada Casa.