

Exercicis Programació Orientada a Objectes

1.

Realitza una classe *Temperatura*, la qual converteixi graus Celsius a Fahrenheit i a l'inrevés. Per això crea dos mètodes *double celsiusAFahrenheit(double)* i *double fahrenheitACelsius(double)*.

Cal que tinguis en compte les següents fórmules que relacionen les dues temperatures:

$$C = (F-32)/1,8$$

$$F = 1,8C + 32$$

2.

Donada la següent classe *cotxe*:

```
class cotxe{
    private int velocitat;
    cotxe () {velocitat=0;}
}
```

Afegeix a la classe *cotxe* els següents mètodes:

- *int getVelocitat()*. Aquest mètode ha de retornar la velocitat actual.
- *void accelera(int mes)*. Aquest mètode actualitza la velocitat a mes quilòmetres més.
- *void frena(int menys)*. Aquest mètode actualitza la velocitat a menys quilòmetres menys.

3.

Realitza una classe *conversor* que converteixi dòlars a euros i a l'inrevés. Codifica els mètodes *dolarsAEuros* i *eurosADolars*.

La classe hauria de tenir dos constructors:

- Un constructor *conversor()* que establirà el valor Euro-Dòlar a 1,2732 (1 € = 1,2732 \$)
- Un constructor *conversor(double)*, el qual permetrà configurar el canvi Euro-Dòlar.

4.

Realitza una classe *nombre* que emmagatzemi un nombre enter i tingui les següents característiques:

- Un constructor per defecte que inicialitzi a 0 el nombre intern
- Un constructor que inicialitzi el nombre intern
- Mètode *suma* que permeti sumar un nombre al valor intern.
- Mètode *resta* que resti un nombre al valor intern.
- Mètode *getValor* que retorni el valor intern.
- Mètode *getDoble* que retorni el doble del valor intern.
- Mètode *getTriple* que retorni el triple del valor intern.
- Mètode *setNombre* que inicialitzi de nou el valor intern.

5.

Corregeix els problemes de compilació de la següent classe:

```
/*Classe satelit*/
public class satelit{
    private double meridia;
    private double paral·lel
    private double distancia_terra;

    satelit (double m, double p, double d){
        meridia=m;
        paral·lel=p;
        distancia_terra=d;
    }

    satelit (){
        meridia=paral·lel=distancia_terra=0;
    }

    public void setPosicio(double m; double p; double d){
        meridia=m;
        paral·lel=p;
        distancia_tierra=d;
    }

    public void printPosicio(){
        System.out.println('El satèlit es troba en el paral·lel " +
        paral·lel+" Meridià " + meridia + " a una distància de la terra de " +
        distancia_terra + " Quilòmetres");
    }
}
```

6.

Modifica l'exercici anterior i afegeix el següents mètodes:

- Mètode *void variaAlsada(double desplaçament)* Aquest mètode accepta un paràmetre que serà positiu o negatiu depenent de si el satèl·lit ha d'allunyar-se o aproximar-se a la Terra.
- Mètode *boolean enOrbita()* Aquest mètode tornarà *false* si el satèl·lit està en terra i *true* en cas contrari.
- Mètode *void variaPosicio(double variap, double variam)*. Aquest mètode permet modificar els atributs de posició (meridià i paral·lel) mitjançant els paràmetres *variap* i *variame* respectivament. Aquests paràmetres seran valors positius o negatius relatius que faran que el satèl·lit modifiqui la seva posició.

7.

Crea la classe *pes*, la qual tindrà les següents característiques:

- Haurà de tenir un atribut en què s'emmagatzemi el pes d'un objecte en quilograms.
- En el constructor se li passarà el pes i la mida en que s'ha pres ('Lb' per lliures, 'Li' per lingots, 'Oz' per onces, 'P' per a penics, 'K' per a quilos, 'G' per grams i 'Q' per quintals.

- Haurà de tenir els següents mètodes:
 - `getLliures`. Torna el pes en lliures
 - `getLingots`. Torna el pes en lingots
 - `getPes`. Torna el pes en la mida que es passa com a paràmetres (mateixa codificació que abans)
- Per realitzar l'exercici pren com a referència les següents dades:
 - 1 Lliura = 16 onces = 453 grams.
 - 1 Lingot = 32,17 lliures = 14,59 kg
 - 1 Onça = 0,0625 lliures = 28,35 grams
 - 1 Penic = 0,05 onces = 1,55 grams
 - 1 Quintal = 100 lliures = 43,3 kg

8.

Crea una classe amb un mètode *millesAMetres* que pren com a paràmetre d'entrada un valor en milles marines i les converteix a metres.

Una vegada tinguis aquest mètode, escriu un altre *millesAQuilometres* que fa la mateixa conversió, però aquesta vegada expressa el resultat en Quilòmetres

- Nota: 1 milla marina equival a 1852 metres.

9.

Crea la classe *cotxe* amb dos constructors. Un no prendrà cap paràmetre i l'altre sí. Els dos constructors inicialitzaran els atributs *marca* i *model* de la classe (inventa't els valors per defecte de la marca i del model).

10.

Implementa una classe *consum*, la qual forma part de la centralita electrònica d'un cotxe i té les següents característiques:

- Atributs:
 - *kms*. Quilòmetres recorreguts pel cotxe
 - *litres*. Litres de combustible consumits
 - *vmed*. Velocitat mitjana
 - *pcom*. Preu del combustible
- Mètodes:
 - `getTemps`. Indicarà el temps emprat en realitzar el viatge
 - `consumMitja`. Consum mitjà del vehicle (en litres cada 100 quilòmetres)
 - `consumEuros`. Consum mitjà del vehicle (en euros cada 100 quilòmetres)

No oblidis de crear un constructor per a la classe que estableixi el valor dels atributs. Tria el tipus de dada més adient per a cada atribut.

11.

Per a la classe anterior implementa els següents mètodes, els quals podran modificar els valors dels atributs de la classe:

- setKms
- setLitres
- setVmed
- setPcom

12.

El restaurant mexicà d'Israel l'especialitat del qual són les papes amb sípia ens demana dissenyar un mètode amb el que es pugui saber quants clients poden atendre amb la matèria primera que tenen en el magatzem. El mètode rep la quantitat de papes i sípies en quilos i retorna el nombre de clients que pot atendre el restaurant tenint en compte que per cada tres persones, Israel fa servir un quilo de papes i mig de sípia.

12 + 1.

Modifica el programa anterior creant una classe que permeti emmagatzemar els quilos de papes i sípies del restaurant. Implementa els següents mètodes:

- *public void addSipies(int x)*: Afegeix x quilos de sípia als ja existents.
- *public void addPapes(int x)*: Afegeix x quilos de papes als ja existents.
- *public int getComensals()*. Torna el nombre de clients que pot atendre el restaurant (aquest és el mètode anterior)
- *public void showSipies()*. Mostra per pantalla els quilos de sípia que hi ha al magatzem
- *public void showPapes()*. Mostra per pantalla els quilos de papes que hi ha al magatzem.