Unidad 3.- Programación Orientada a Objetos. Programación JAVA.

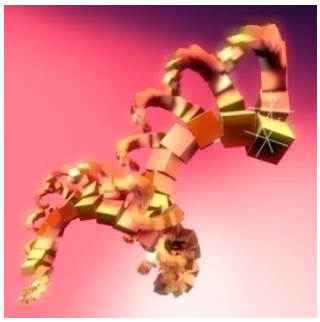
Desarrolla un proyecto JAVA llamado nombre_examen_Asir que contenga los siguientes elementos:

- Un paquete Ejercicio1_Examen.
- Un paquete Ejercicio2_Examen.
- Un paquete Ejercicio3_Examen.
- Una clase Main.java que se utilizará para probar de manera opcional los ejercicios resueltos. Es evidente que el examen estará dividido en tres ejercicios y cada una de ellos se resolverá con una , dos ..etc clases que serán deficinas en su correspondiente paquete.
- *Se reserva un punto para la calidad del proyecto*.

Ejercicios 1 (3 puntos).- (Paquete Ejercicio1_Examen)

Clase ArrayAsi

/ clase que moldea y contiene las operaciones relacionadas con un vector de enteros (int v[]) */*



Atributos *(el alumno decidirá si un atri. es público o privado)*:

int elementos [];

int num_elemente;

Métodos (cada método 0,75p):

public ArrayAsi(int num);

/* Contructor que inicializa cada uno de los atributos. El parámetro indica el tamaño del vector de enteros. */

public int min_ArrayAsi();

/* Método que devuelve el menor de los enteros contenido en el atributo elementos. */

public static ArrayAsi suma(ArrayAsi a, ArrayAsi b);

/* Método estático que proporciona la utilidad para sumar dos ArrayAsi *(suma de vectores).* Si la longitud de ambos no coincide devolveré **null** y sino realizaré la suma. */

public void ordenar_ArrayAsi();

/* Método que ordena los enteros contenidos en el atributo elementos. */

Ejercicios 2 (3 puntos).- (Paquete Ejercicio2_Examen)

Clase Radio

/ Se quiere definir una clase que permita controlar un sintonizador digital de emisoras*/*

Atributos ***(el alumno decidirá si un atri. es público o privado, y podrá añadir otros para solucionar el problema)***:

double frecuencia;

int volumen;

String canal;

Métodos (cada método 0,5p):

public void up_hez()
/* Método para subir (up) la frecuencia ***(en saltos de X MHz entre [X1,X2])***. *Deberá controlarse los límites. */*
public void down_vol()
/* Método para bajar (down) el volumen ***(en saltos de Y dB entre [Y1,Y2])***. *Deberá controlarse los límites.*
public void displayRadio()
/* Método para mostrar los valores de la clase */
public void changeCanal()
/* Método que cambia entre FM/AM. El cambio situará los atributos a valores por defecto. */
public void reset()
/* Método que pone los atributos a valores por defecto */
public Radio()
El constructor por defecto inicializará la emisora a valores por defecto en el canal FM.

Nota: Por defecto los valores son:

a) ***canal="FM"***
   frecuencia=30
   volume=0
b) ***canal="AM"***
   frecuencia=10
   volume=3

Nota:

a) FM: ***(saltos de 5 MHz entre [30,80]) y (saltos de 5 dB entre [0,20])***
b) AM: ***(saltos de 3 MHz entre [10,40]) y (saltos de 2 dB entre [3,18])***

Ejercicios 3 (3 puntos).- (Paquete Ejercicio3_Examen)

Clase Triangulo

/ Se quiere definir una clase que represente al polígono Triangulo. */*

Atributos *(el alumno decidirá si un atri. es público o privado)*:

double long_lado1;

double long_lado2;

double long_lado3;

Métodos (cada método 0,75p):

public Triangulo (double l1, double l1, double l1)

/* Constructor sobre cargado que inicializa los atributos. */

public boolean compareTo_Triangulos(Triangulo a, Triangulo b)

/* Método que que permite determinar si dos triangulos son o no iguales. */

public boolean compareTo_VTriangulos(Triangulo v[])

/* Método que permite determinar si un conjunto de Triangulos son iguales. */

public int tipo_Triangulo()

/* Método que indica si el triangulo es equilátero (1), isósceles (2), escaleno (3). */

Por la longitud de sus lados, los triángulos se clasifican en:

- ***Triángulo equilátero**: si sus tres lados tienen la misma longitud.*
- ***Triángulo isósceles**: si tiene dos lados de la misma longitud.*
- ***Triángulo escaleno**: si todos sus lados tienen longitudes diferentes.*