

1º.- Desarrollar una aplicación de control de llamadas realizadas en una centralita telefónica. La centralita ira registrando el número de llamadas realizadas así como el coste total de todas las llamadas realizadas.

Las llamadas tienen como atributos el número de origen, el número de destino y su duración en segundos.

Existen dos tipos de llamadas:

- Llamadas locales que cuestan 15 céntimos el segundo.
- Llamadas provinciales que dependiendo de la franja horaria en la que se realicen cuestan: 20 céntimos/sg en franja 1, 25 céntimos/sg en franja 2 y 30 céntimos/sg en franja 3.

Finalmente, desarrollar la clase principal que cree una centralita, y registre las llamadas de distinto tipo que vienen en el fichero de texto "llamadas.txt" y le pida a la centralita que muestre por pantalla las llamadas según las vaya registrando, e informe del número total de llamadas de cada tipo y la facturación total realizada.

3º.- Crear la clase Punto, cuyos atributos son:

int coordX, int coordY, double distOrigen

y que tendrá los constructores y métodos necesarios para el buen funcionamiento de dicha clase. La fórmula para calcular la distancia al origen es:

$$\text{distOrigen} = \sqrt{\text{coordX}^2 + \text{coordY}^2}$$

Posteriormente, añadir a dicha clase lo siguiente:

a) Un método estático de nombre almacenarPuntosEnArchivo que guarde en un fichero serializable los diferentes puntos que se crearan a partir de los valores almacenados en el fichero binario puntos.bin. La estructura del fichero binario es:

valorX int , valorY int que se corresponden a las coordenadas de los puntos

El método recibirá por parámetro el nombre de ambos ficheros. Se debe tener en cuenta que cada vez que se llame al método los puntos se deben añadir al fichero, sin perder los que ya se tengan previamente.

b) Añadir otro método estático, de nombre obtenerArrayListDeArchivo, que reciba por parámetro el nombre del archivo que contiene los objetos de la clase Punto y devuelva un ArrayList con los datos de dicho archivo..

Finalmente, hacer la clase principal que compruebe funcionamiento de los métodos de los dos apartados anteriores.
