

PROGRAMACION

Centralita telefónica



MARZO 2019

PROGRAMACION

Indice

EJERCICIO DE PROGRAMACION	3
PROPUESTA DE CODIGO	4
UML	6

PROGRAMACION

EJERCICIO DE PROGRAMACION

[CENTRALITA TELEFÓNICA]

Desarrollar una aplicación de control de llamadas realizadas en una centralita telefónica.

En la centralita se van registrando llamadas. Registrar una llamada consiste en contabilizar el número de llamadas realizadas así como el coste total de todas las llamadas realizadas.

La centralita mostrará por pantalla todas las llamadas según las vaya registrando.

Existen dos tipos de llamadas:

Las llamadas locales que cuestan 15 céntimos el segundo.

Y las llamadas provinciales que dependiendo de la franja horaria en la que se realicen cuestan: 20 céntimos en franja 1, 25 céntimos en franja 2 y 30 céntimos en franja 3, cada segundo.

Todas las llamadas tienen como datos el número origen de la llamada, el número destino y su duración en segundos.

Desarrollar la clase Practica7a que en su método main cree una centralita, registre varias llamadas de distinto tipo y le pida a la centralita un informe con el número total de llamadas y la facturación total realizada.

Dibujar Diagrama de Clases UML.

PROGRAMACION

Propuesta deCodigo.

(Clase Centralitatelefonica)

```
public class Centralitatelefonica {  
  
    private int nLLamadas;  
    private double costeTotal;  
  
    public Centralitatelefonica(){  
        nLLamadas=0;  
        costeTotal=0;  
    }  
  
    public int getNLLamadas(){  
        return nLLamadas;  
    }  
  
    public double getCosteLLamadas(){  
        return costeTotal;  
    }  
  
    public void registraLLamada(LLamada llamada){  
        nLLamadas++;  
        costeTotal+=llamada.costeLLamada();  
    }  
  
    public void printInforme (){  
        System.out.println("El número total de llamadas que realizaste es "+nLLamadas+" \n El coste total de la Factura es "+costeTotal);  
    }  
}
```

(Clase LLamada)

```
public abstract class LLamada {  
  
    private String nOrigen;  
    private String nDestino;  
    private int duracion;  
  
    public LLamada(String nor, String ndes, int dur) {  
        // TODO Auto-generated constructor stub  
        nOrigen=nor;  
        nDestino=ndes;  
        duracion=dur;  
    }  
  
    public int getDuracion(){  
        return duracion;  
    }  
    public abstract double costeLLamada();  
}
```

PROGRAMACION

Propuesta deCodigo.

(Clase LLamadasLocales)

```
public class LLamadasLocales extends LLamada {  
  
    private double coste;  
  
    public LLamadasLocales(String nor,String ndes,int dur){  
        super(nor,ndes,dur);  
        coste=0.15;  
    }  
  
    public double costeLLamada(){  
        double costetotal=coste*super.getDuracion();  
        return costetotal;  
    }  
}
```

(Clase LLamadaProvinciales)

```
public class LLamadasProvinciales extends LLamada {  
  
    private double coste=0;  
    private int franja;  
  
    public LLamadasProvinciales(String nor,String ndes,int dur, int f){  
        super(nor,ndes,dur);  
        franja=f;  
        switch (franja){  
            case 1: coste=0.20;break;  
            case 2: coste=0.25;break;  
            case 3: coste=0.30;break;  
        }  
    }  
  
    public double costeLLamada(){  
        double costetotal=coste*super.getDuracion();  
        return costetotal;  
    }  
}
```

(Clase Main)

```
public class Practica7a {  
  
    public static void main(String[] args) {  
        Centralitatelefonica centralitatelefonica=new Centralitatelefonica();  
        //llamadas locales  
        LLamadasLocales llamadalocal1=new LLamadasLocales("666156599","661516998",10);  
        LLamadasLocales llamadalocal2=new LLamadasLocales("665595159","615984126",10);  
        LLamadasLocales llamadalocal3=new LLamadasLocales("661561123","616551165",10);  
        LLamadasLocales llamadalocal4=new LLamadasLocales("665595159","615984126",10);  
        LLamadasLocales llamadalocal5=new LLamadasLocales("661561123","616551165",10);  
        //llamadas provinciales  
        LLamadasProvinciales llamadaprov1=new LLamadasProvinciales("156891591","151589596",3,1);  
        LLamadasProvinciales llamadaprov2=new LLamadasProvinciales("198915169","981595995",2,2);  
        LLamadasProvinciales llamadaprov3=new LLamadasProvinciales("915619599","898156599",5,3);  
        LLamadasProvinciales llamadaprov4=new LLamadasProvinciales("198915169","981595995",2,2);  
        LLamadasProvinciales llamadaprov5=new LLamadasProvinciales("915619599","898156599",5,3);  
        //llamada al metodo de la clase centralitatelefonica  
        centralitatelefonica.registraLLamada(llamadlocal1);  
        centralitatelefonica.registraLLamada(llamadaprov1);  
        centralitatelefonica.registraLLamada(llamadlocal2);  
        centralitatelefonica.registraLLamada(llamadaprov2);  
        centralitatelefonica.registraLLamada(llamadlocal3);  
        centralitatelefonica.registraLLamada(llamadaprov3);  
        centralitatelefonica.registraLLamada(llamadlocal4);  
        centralitatelefonica.registraLLamada(llamadaprov4);  
        centralitatelefonica.registraLLamada(llamadlocal5);  
        centralitatelefonica.registraLLamada(llamadaprov5);  
        //impresion por pantalla numero de llamadas y factura  
        centralitatelefonica.printInforme();  
    }  
}
```

PROGRAMACION

UML

Diseño de Diagrama de Clases UML.

