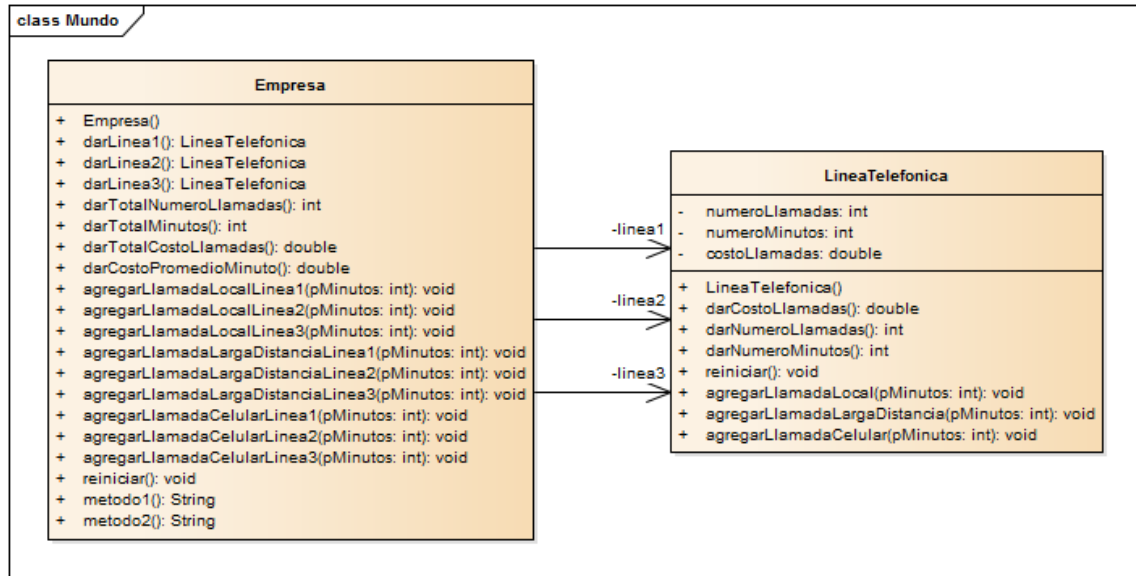


Sección 1 – Manejo de atributos, inicialización y manejo de métodos. (Hacer en hojas blancas).

Considerando el ejercicio de las líneas telefónicas, cuyo diagrama UML del modelo del mundo se muestra a continuación, desarrolle en el espacio indicado los siguientes puntos.



La empresa quiere empezar a hacer un control sobre las llamadas a celular. Para esto, necesita conocer sobre las líneas telefónicas, el número de minutos consumidos en llamadas a celular.

1. Agregue a la clase **LineaTelefonica** un nuevo atributo llamado **numeroMinutosCelular** para guardar la cantidad de minutos consumidos en llamadas a celular.

```

public class LineaTelefonica
{
    ...
    //-----
    // Atributos
    //-----

    /**
     * Número de minutos consumidos en llamadas a celular
     */

    ...
}
  
```

2. Escriba el **método constructor** de la clase **LineaTelefonica** que asigna el valor cero a todos sus atributos.

```
public class LineaTelefonica
{
    ...

    /**
     * Inicializa en cero el número de llamadas, el número de minutos, el número
     * de minutos a celular y el costo de las llamadas
     */

    public LineaTelefonica( )

    {

    }
}
```

3. Escriba el **método constructor** de la clase **Empresa** que inicializa las 3 líneas telefónicas.

```
public class Empresa
{
    ...

    /**
     * Inicializa las 3 líneas telefónicas
     */

    public Empresa()

    {

    }
}
```

4. Complete el método **darNumeroMinutosCelular** de la clase **LineaTelefonica** que devuelva (retorne) el número de minutos consumidos en llamadas a celular sobre la línea

```
public class LineaTelefonica
{
    ...

    //-----
    // Métodos
    //-----

    /**
     * Devuelve el número de minutos consumidos en llamadas a celular
     */

    public int darNumeroMinutosCelular ( )
    {

    }
}
```

5. Construya el método **darTotalMinutosCelular** de la clase **Empresa** que retorne el número total de minutos consumidos en llamadas a celular realizadas en las 3 líneas telefónicas. Es decir, la suma de los minutos de llamadas a celular realizadas en cada línea telefónica. El nuevo método **darTotalMinutosCelular** debe utilizar (llamar) **obligatoriamente** el método **darNumeroMinutosCelular ()** (que usted construyó en el punto 2) sobre linea1, linea2 y linea3, para obtener de cada una de ellas, el número de minutos consumidos en llamadas a celular.

```
public class Empresa
{
    ...

    //-----
    // Métodos
    //-----

    /**
     * Devuelve el número total de llamadas realizadas a celular
     */

    public int darTotalMinutosCelular ( )
    {

    }
}
```

6. La empresa necesita conocer qué porcentaje representa la cantidad total de minutos de llamadas a celular con respecto al total de minutos consumidos en todos los tipos de llamadas en todas las líneas. Construya el método **darPorcentajeCelular** de la clase **Empresa** que calcule y retorne este valor. Este método debe utilizar **obligatoriamente** los métodos **darTotalMinutos()** y **darTotalMinutosCelular()** (que usted construyó en el punto 4) de la clase **Empresa** para calcular este valor.

```
public class Empresa
{
    ...

    // -----
    // Métodos
    // -----

    public double darPorcentajeCelular( ){

    }

    ...
}
```

Sección 2 - Manejo de atributos, inicialización y manejo de métodos. (Realizar en el ejemplo Elecciones)

1. Descargue el ejemplo **Elecciones** de la página de Cupi2: <http://cupi2.uniandes.edu.co/sitio/index.php/cursos/apo1/nivel-1>
2. Descomprímalo y ábralo en Eclipse.
3. Lea el archivo **Descripción.doc**, este archivo está ubicado en **n1_elecciones/docs/specs**.
4. Revise el archivo **Modelo Conceptual.jpg**, este archivo está ubicado en **n1_elecciones/docs/specs**.

Los organizadores de la elección del presidente de Cupi2 quieren adicionar una funcionalidad para conocer la cantidad de colaboradores que tiene cada uno de los candidatos. Realice los cambios necesarios para agregar esta funcionalidad:

5. Para esto agregue un atributo en la clase **Candidato** con el nombre **numeroColaboradores** de tipo **int**. Inicialícelo en **0**.
6. Cree el método **darNumeroColaboradores()** que devuelva el número de colaboradores que tiene el candidato.
7. Cree el método **aumentarNumeroColaboradores()** que agregue un colaborador al candidato.
8. En la clase **Urna** agregue un método que se llame **darNumeroColaboradoresCandidato1()**, otro que se llame **darNumeroColaboradoresCandidato2()** y uno que se llame

darNumeroColaboradoresCandidato3(). Estos métodos deben retornar el número de colaboradores que tiene cada candidato.

9. Agregue tres métodos, uno que se llame **aumentarNumeroColaboradoresCandidato1()**, otro que se llame **aumentarNumeroColaboradoresCandidato2()** y uno que se llame **aumentarNumeroColaboradoresCandidato3()**. Estos métodos deben aumentar la cantidad de colaboradores de cada uno de los candidatos.
10. Para probar su funcionamiento modifique el **metodo1()** para que aumente en 1 el número de colaboradores del candidato 1 y retorne **"aumenté bien"**. Modifique el **metodo2()** para que coloque en el retorno el número de colaboradores del candidato 1 (Para esto escriba **return ""+metodoAEjecutar();**). Ejecute el programa, al oprimir el botón de **Opción 1** debe aumentar el número de colaboradores del candidato 1, al oprimir el botón de **Opción 2** debe retornar el número de colaboradores del candidato 1.
11. Haga los cambios correspondientes en el **metodo1()** y en el **metodo2()** para comprobar el funcionamiento de la nueva funcionalidad con el candidato 2 y el candidato 3