

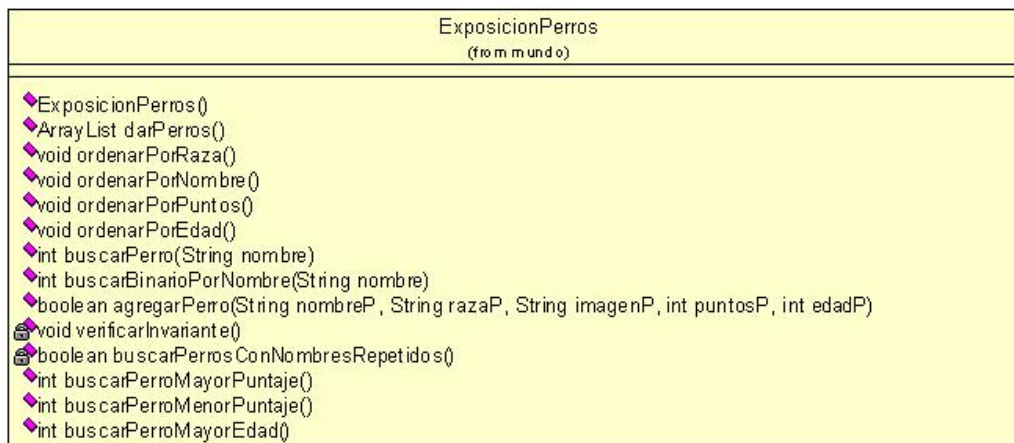


**Universidad de los Andes**  
Ingeniería de Sistemas y Computación  
ISIS1205 – Algorítmica y Programación 2  
Hoja de trabajo teórica Nivel 7 (N7-EE)

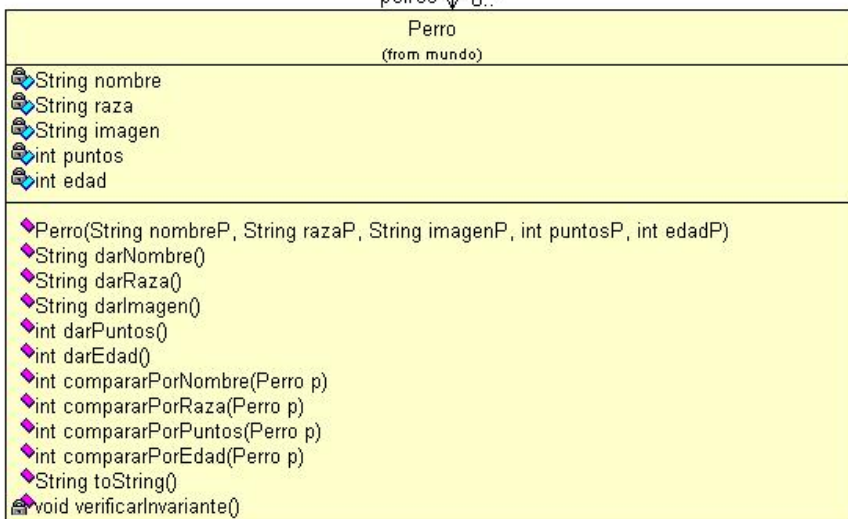


**Enunciado:**

Desarrolle las modificaciones al programa exposiciónCanina según los pasos nombrados a continuación. Apóyese en el siguiente diagrama de clases para saber qué métodos puede usar.



**ExposicionPerros**  
Es la clase que se encarga de manejar, organizar, cargar y salvar los perros  
inv: perros != null y no hay dos perros con el mismo nombre



**Perro**  
Es la clase que representa a un perro  
inv: puntos >= 0 y edad > 0 y imagen != null y nombre != null y raza != null

1. [5%] Agregue un atributo de tipo Entero en la clase Perro que represente la cantidad de hijos que tiene el perro. Por restricción del concurso, solo podrán entrar a concursar perros que tengan más de 2 hijos, y cuyo número de hijos sea par.

**public class** Perro

```
{
```

```
...
```

```
...
```

```
}
```

2. [5%] Agregue un método para dar el número de hijos del perro.

**public class** Perro

```
{
```

```
...
```

...

}

3. [10%] Complemente el método de verificarinvariante agregando los nuevos assert para los cambios realizados en el mundo. (hint: recuerde que para saber si es par o no se puede utilizar la instrucción '%' )

**public class** Perro

{

...

private void verificarInvariante( )

{

assert ( puntos >= 0 ) : "Los puntos no pueden ser menores a 0";

assert ( edad > 0 ) : "La edad no puede ser 0";

assert ( imagen != null ) : "La imagen no puede ser null";

assert ( nombre != null ) : "El nombre no puede ser null";

assert ( raza != null ) : "La raza no puede ser null";

...

```
}
```

```
...
```

```
}
```

4. [10%] En la clase Perro Agregue un método para comparar contra otra Perro dada su numero de hijos. **El método retorna 0 si son iguales, 1 si es menor el recibido por parámetros y -1 si el recibido por parámetro es mayor**

```
public class Perro
```

```
{
```

```
...
```

```
public int compararPorNumeroHijos( Perro elPerro )
```

```
{
```

```
}
```

```
...
```

```
}
```

5. [20%] En la clase `ExposicionCanina` añada un método que busque un Perro según su cantidad de Hijos (retorna el primer perro que encuentre o null en caso de no encontrar ninguno). Escriba el método suponiendo que el vector de Perros está ordenado por número de hijos DESCENDENTEMENTE. Utilice el método de **BÚSQUEDA BINARIA**

```
public class ExposicionCanina
```

```
{
```

```
...
```

```
public Perro buscarBinarioPorNumeroHijos( int cantHijos )
```

```
{
```

```
}
```

```
...
```

```
}
```

6. [30%]Escriba en la clase `ExposicionCanina` un método que ordene a las Perros **DESCENDENTEMENTE (de mayor a menor)** por su número de hijos utilizando el algoritmo de selección.

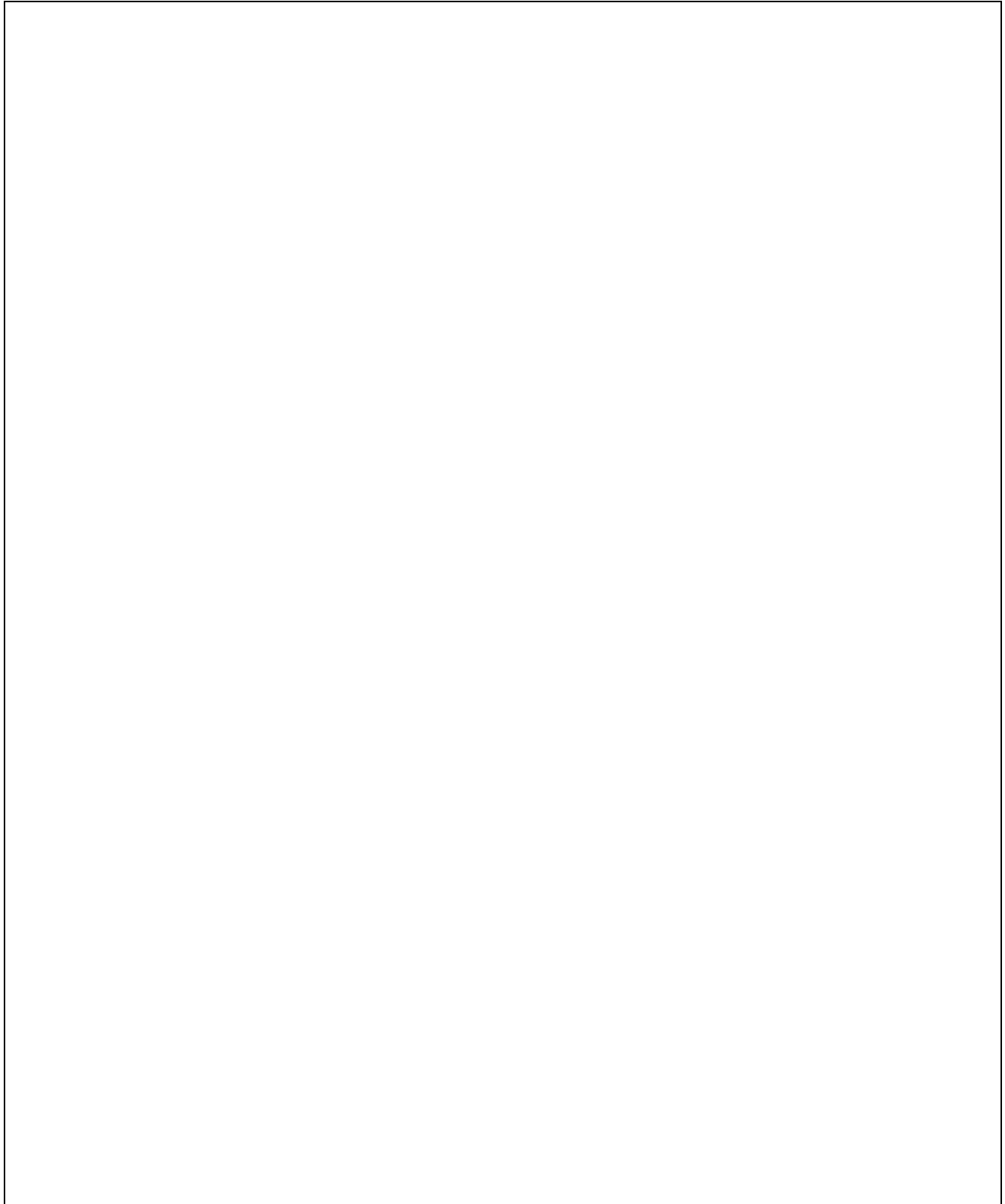
```
public class ExposicionCanina
```

```
{
```

```
...
```

```
public void ordenarPorNumeroHijos( )
```

```
{
```



```
}
```

```
...
```

```
}
```

7. [20%] Dado el siguiente Escenario Complete la clase ExposicionCaninaTest.

```
import junit.framework.TestCase;

import uniandes.cupi2. exposicionCanina.mundo.Perro;

import uniandes.cupi2. exposicionCanina.mundo.ExposicionCanina;

public class ExposicionCaninaTest extends TestCase

{

    private ExposicionPerros exposicion;
```



```
private int cantidadPerros;
```

```
private void setupEscenario1( )
```

```
{
```

```
    exposicion = new ExposicionPerros( );
```

```
    cargarPerros( "./test/data/perros1.txt" );
```

```
}
```

**\*\* Este método se encarga de verificar el método ordenarPorNumeroHijos <br>**

**\* <b> Objetivo: </b> Probar que se ordenen las Perros de la ExposicionCanina por su número de hijos <br>**

**\* <b> Resultados esperados: </b> <br>**

**\* 1. La cantidad de Perros es la misma después de ordenadas <br>**

**\* 2. Al ordenar, el Perro con menor cantidad de hijos quedar de ultimo y el de mayor de primero.<br>**

**\* Para cualquier elemento su anterior debe tener un numero de hijos mayor**

**\*/**

```
public void testOrdenarPorNumeroHijos ( )
```

```
{
```

```
    setupEscenario1( );
```

```
    ArrayList Perros = exposicion.darPerros( );
```

}  
}