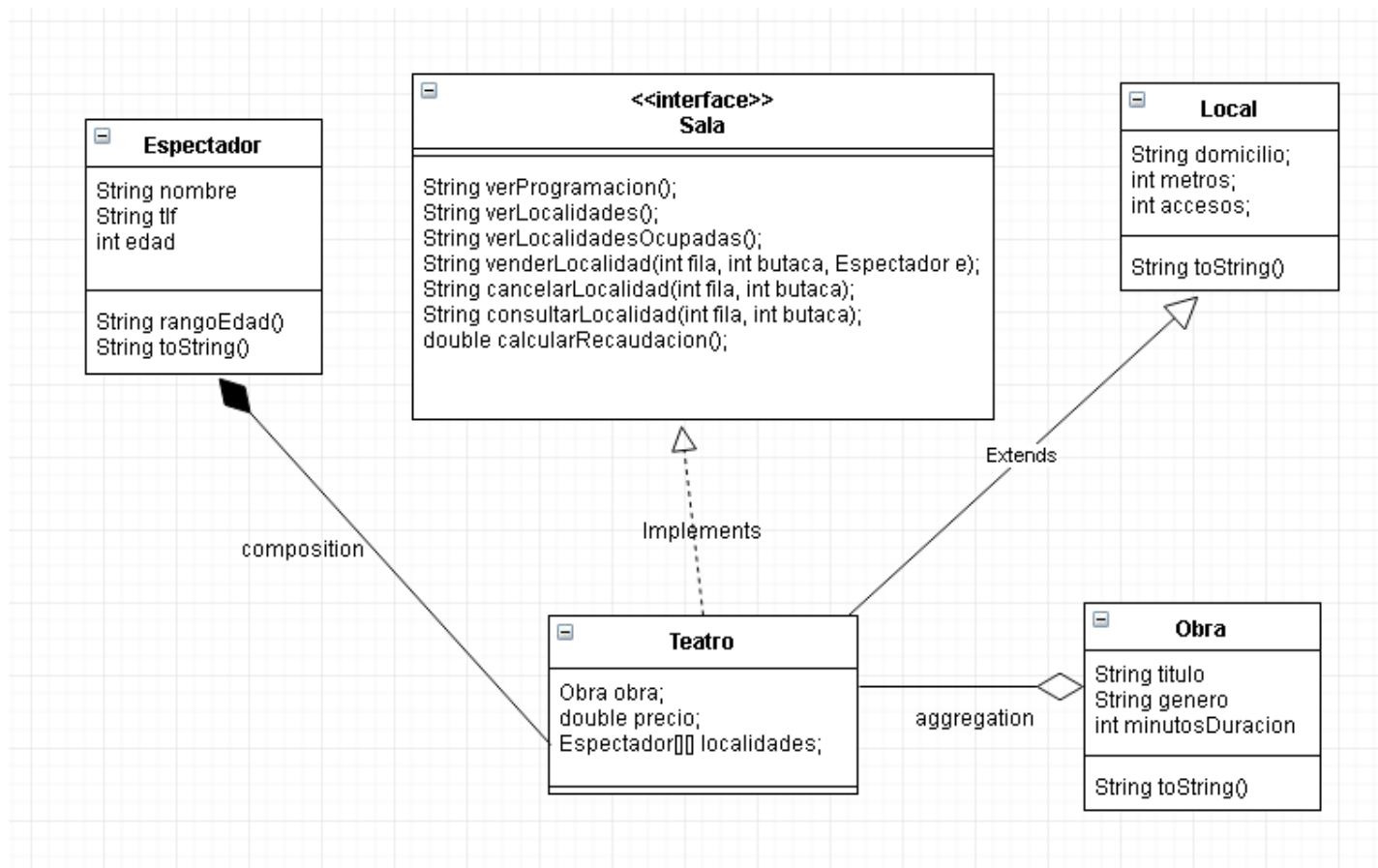


Enunciado

Especificaciones para el ejercicio (ESTA ACTIVIDAD NO ES OBLIGATORIA NI COMPUTA)

El trabajo práctico consiste en la realización de una aplicación para la gestión de un pequeño teatro. Para lograrlo tendréis que implementar este modelo de clases:



Para implementar en Java toda esta jerarquía de clases debes tener en cuenta las siguientes cosas:

- La clase **Teatro** implementa la interfaz **Sala**, luego está obligada a implementar todos sus métodos abstractos, cuya cabecera ves claramente en el diagrama. Esta interfaz dicta la norma de cómo deben comportarse aquellas clases que modelan salas de representación de cualquier tipo (cine, teatro, magia, etc.).
- La clase **Teatro** extiende o hereda de la clase **Local** y de ella coge las propiedades que definen el tipo de local donde está ubicado el teatro.
- Un objeto **Teatro** está compuesto por una matriz (array de 2 dimensiones) de objetos **Espectador**, personas que se sentarán en una determinada butaca para ver una obra. El teatro tiene 5 filas (0 a 4) y 10 butacas (0 a 9), representadas por las filas y columnas de la matriz. Cada celda representa una butaca, que contiene una referencia al objeto **Espectador** que reservó la butaca, o *null* si la butaca está libre.
- El **Teatro** agrega un objeto **Obra**, que representa la obra de teatro que está en ese momento en la programación.
- Debéis declarar todas las propiedades de todas las clases como privadas y luego hacerlas accesibles mediante métodos *get/set*.

- Todos los métodos serán públicos.
- Cada clase contará con un constructor con los parámetros necesarios para asignar valores a las propiedades. La excepción será el *array* de *localidades*, que no pasará como parámetro al constructor, sino que se creará vacío con una sentencia como esta:

```
this.localidades = new Espectador[5][10];
```

Será a través del método *venderLocalidad* como vaya llenándose.

- Tendréis que descubrir qué código insertar en cada uno de los métodos según los requerimientos del proyecto que os iremos exponiendo a continuación.
- Todo el modelo de clases se creará en el paquete **com.itt.arte**.

Debéis partir del siguiente código para la clase *Principal*:

```
import java.util.Scanner;

import com.itt.arte.*;

public class Principal {

    private static Scanner lector;

    private static Obra obra =

        new Obra("La cena de los idiotas", "Comedia", 95);

    private static Teatro teatro = new Teatro("C/ Sol, 45", 300, 2, obra, 30);

    public static void main(String[] args) {

        lector = new Scanner(System.in);

        char opc;

        do {

            opc = mostrarMenu();

            switch (opc) {

                case '1':

                case '2':

                case '3':

                case '4':

                case '5':

                case '6':

                case '7':

                case '8':

                default:
```

```

        }

        } while (opc!='8');

        lector.close();

    }

    public static char mostrarMenu() {

        String opcion;

        System.out.println ("TEATRO LA BOMBILLA DE DON BLAS");

        System.out.println ("-----");

        System.out.println ("1. Ver la programación actual");

        System.out.println ("2. Mostrar todas las localidades");

        System.out.println ("3. Mostrar localidades ocupadas");

        System.out.println ("4. Vender localidad");

        System.out.println ("5. Cancelar localidad");

        System.out.println ("6. Consultar localidad");

        System.out.println ("7. Calcular recaudación");

        System.out.println ("8. Terminar programa");

        System.out.println ("-----");

        System.out.println ("¿Qué opción deseas?");

        opcion = lector.nextLine();

        return opcion.charAt(0); // Devuelvo el primer caracter tecleado.

    }

}

```

Con este código ya tenéis un menú que os brinda la interfaz de usuario para la ejecución de cada una de las tareas del proyecto. El resultado es el siguiente:

```

Principal (10) [Java Application] C:\Program Files\Java\
TEATRO LA BOMBILLA DE DON BLAS
-----
1. Ver la programación actual
2. Mostrar todas las localidades
3. Mostrar localidades ocupadas
4. Vender localidad
5. Cancelar localidad
6. Consultar localidad
7. Calcular recaudación
8. Terminar programa
-----
¿Qué opción deseas?

```

Al elegir cualquiera de las opciones volverá de nuevo al menú, a no ser que hayáis elegido la opción 8.

Para cada uno de los casos, en la instrucción *switch* debéis invocar a una función independiente, dentro de la clase *Principal*, que realice la tarea específica. Observar que a nivel global de clase se han declarado los objetos *obra* y *teatro*, de este modo los tendréis disponibles desde cualquiera de las funciones. En cada una de las funciones independientes debéis invocar al método que corresponda de la clase *Teatro*, pero antes debéis de solicitar por teclado los datos necesarios para realizar la tarea.

A continuación, os exponemos un ejemplo de ejecución de cada una de las opciones para que podáis decidir qué código necesitáis insertar en cada una de las opciones de menú:

1. Ver la programación actual

```
TEATRO LA BOMBILLA DE DON BLAS
-----
1. Ver la programación actual
2. Mostrar todas las localidades
3. Mostrar localidades ocupadas
4. Vender localidad
5. Cancelar localidad
6. Consultar localidad
7. Calcular recaudación
8. Terminar programa
```

¿Qué opción deseas?

1

Hoy representamos La cena de los idiotas, género: Comedia, Duración: 95
En C/ Sol, 45, local de 300 metros, con 2 accesos
Precio: 30.0

2. Mostrar todas las localidades

```
TEATRO LA BOMBILLA DE DON BLAS
-----
1. Ver la programación actual
2. Mostrar todas las localidades
3. Mostrar localidades ocupadas
4. Vender localidad
5. Cancelar localidad
6. Consultar localidad
7. Calcular recaudación
8. Terminar programa
```

¿Qué opción deseas?

2

0.0 Ocupada	0.1 Libre	0.2 Libre	0.3 Libre	0.4 Libre	0.5 Libre	0.6 Libre	0.7 Libre	0.8 Libre	0.9 Libre
1.0 Libre	1.1 Libre	1.2 Ocupada	1.3 Libre	1.4 Libre	1.5 Libre	1.6 Libre	1.7 Libre	1.8 Libre	1.9 Libre
2.0 Libre	2.1 Libre	2.2 Libre	2.3 Libre	2.4 Libre	2.5 Libre	2.6 Libre	2.7 Libre	2.8 Libre	2.9 Libre
3.0 Libre	3.1 Libre	3.2 Libre	3.3 Libre	3.4 Libre	3.5 Libre	3.6 Libre	3.7 Libre	3.8 Libre	3.9 Libre
4.0 Libre	4.1 Libre	4.2 Libre	4.3 Libre	4.4 Libre	4.5 Libre	4.6 Libre	4.7 Libre	4.8 Libre	4.9 Libre

En la imagen se ve claro que las localidades 0.0 y 1.2 están ocupadas y el resto libres. La localidad 0.0 se refiere a la fila 0 y la butaca 0.

3. Mostrar localidades ocupadas

```
TEATRO LA BOMBILLA DE DON BLAS
-----
1. Ver la programación actual
2. Mostrar todas las localidades
3. Mostrar localidades ocupadas
4. Vender localidad
5. Cancelar localidad
6. Consultar localidad
7. Calcular recaudación
8. Terminar programa
-----
```

¿Qué opción deseas?

3

0.0 CARLOS, tlf: 616664422, Tipo: MAYOR
1.2 ROSA, tlf: 664882222, Tipo: MENOR

Esta opción muestra las localidades ocupadas junto al nombre del cliente, el teléfono y su rango de edad, conseguido a partir del método *rangoEdad()*. Los rangos posibles son:

0 a 12 años: INFANTIL

13 a 17 años: MENOR

18 a 64 años: MAYOR

>= 65: JUBILADO

4. Vender localidad

```
TEATRO LA BOMBILLA DE DON BLAS
-----
1. Ver la programación actual
2. Mostrar todas las localidades
3. Mostrar localidades ocupadas
4. Vender localidad
5. Cancelar localidad
6. Consultar localidad
7. Calcular recaudación
8. Terminar programa
-----
```

```
¿Qué opción deseas?
4
¿En qué fila quieres sentarte (0-4)?
2
¿Y en qué butaca (0-9)?
3
¿Cómo se llama?
CARLA
¿Su teléfono?
699482223
¿Cuántos años tiene?
12
Se ha vendido la localidad 2.3 a CARLA por 15.0 euros
```

Esta opción debe crear un objeto *Espectador* y asignarlo a una localidad (butaca) y debe informar sobre la localidad vendida, a quién y el precio. Al precio de la localidad, registrado en la propiedad *precio*, se le aplica un descuento dependiendo del rango de edad:

INFANTIL 50% descuento

MENOR 20% descuento

MAYOR 0% descuento

JUBILADO 66% descuento

En el caso de intentar vender una localidad que ya está ocupada, el programa debe informar con el mensaje “Esa plaza ya está ocupada” o algo similar.

5. Cancelar localidad

```
TEATRO LA BOMBILLA DE DON BLAS
-----
1. Ver la programación actual
2. Mostrar todas las localidades
3. Mostrar localidades ocupadas
4. Vender localidad
5. Cancelar localidad
6. Consultar localidad
7. Calcular recaudación
8. Terminar programa
```

```
¿Qué opción deseas?
5
¿Fila (0-4)?
2
¿Butaca (0-9)?
3
CARLA ha cancelado su reserva
```

Esta opción libera una localidad asignándole de nuevo el valor *null* a la posición que corresponda de la matriz.

Si se intenta liberar una localidad que ya está libre, el programa debe informar con el mensaje “La plaza ya está libre” o algo similar.

6. Consultar localidad

```
TEATRO LA BOMBILLA DE DON BLAS
-----
1. Ver la programación actual
2. Mostrar todas las localidades
3. Mostrar localidades ocupadas
4. Vender localidad
5. Cancelar localidad
6. Consultar localidad
7. Calcular recaudación
8. Terminar programa
```

```
¿Qué opción deseas?
6
¿Fila (0-4)?
1
¿Butaca (0-9)?
2
Localidad ocupada por ROSA, tlf: 664882222, Tipo: MENOR, Precio: 24.0
```

Esta opción muestra una información más detallada de una reserva de localidad.

7. Calcular recaudación

```
TEATRO LA BOMBILLA DE DON BLAS
-----
1. Ver la programación actual
2. Mostrar todas las localidades
3. Mostrar localidades ocupadas
4. Vender localidad
5. Cancelar localidad
6. Consultar localidad
7. Calcular recaudación
8. Terminar programa
-----
```

```
¿Qué opción deseas?
```

```
7
```

```
Recaudación: 54.0
```

Esta opción nos dará la suma total de los precios de todas las localidades vendidas.

8. Terminar programa

```
TEATRO LA BOMBILLA DE DON BLAS
-----
1. Ver la programación actual
2. Mostrar todas las localidades
3. Mostrar localidades ocupadas
4. Vender localidad
5. Cancelar localidad
6. Consultar localidad
7. Calcular recaudación
8. Terminar programa
-----
```

```
¿Qué opción deseas?
```

```
8
```

```
Hasta pronto
```

Esta opción finaliza el programa mostrando el mensaje “Hasta pronto”.