



Sean las siguientes definiciones de tipos:

Tipo Espacio

Propiedades:

- nombre: String, consultable
- capacidad: Integer, consultable

Tipo Actividad

Propiedades:

- fecha: LocalDate, consultable y modificable
- horaComienzo: LocalTime, consultable
- tipo: del tipo enumerado TipoActividad (REUNION, CONFERENCIA, TALLER, OTRO), consultable
- lugar: Espacio, consultable y modificable
- asistentes: Integer, consultable y modificable
- inscripcion: Boolean, consultable. Toma valor true si la actividad requiere inscripción previa, y false si no la requiere
- maximoAsistentes: Integer, consultable, derivada. El número máximo de asistentes a la actividad viene dado por la capacidad del espacio donde se realiza

Tipo Evento

Propiedades:

- nombre: String, consultable
- actividades: List<Actividad>, consultable

El tipo Espacio está implementado mediante un record, y los tipos Actividad y Evento mediante una clase.

1. La clase Actividad tiene un constructor con un parámetro por cada propiedad básica, en el orden en que están definidas. Escriba las instrucciones que construyen los siguientes objetos:
 - a) Un espacio e con el nombre "A3.11" y la capacidad 175.
 - b) Una actividad a con la fecha actual, la hora 11:30, el tipo OTRO, el lugar igual al espacio e creado antes, 70 asistentes y sin inscripción previa.
2. Dados un objeto e de tipo Espacio y un objeto a de tipo Actividad, escriba una expresión lógica que tome valor true si se cumple cada una de las siguientes condiciones, y false en caso contrario:
 - a) El espacio donde se celebra la actividad a es igual al espacio e
 - b) El nombre del espacio e no está vacío
 - c) El número de asistentes a la actividad a es positivo y menor o igual que el máximo de asistentes
 - d) La fecha de la actividad a es el 12 de marzo de 2024
3. El número de asistentes a una actividad debe ser mayor o igual que 0. Indique en qué lugar o lugares de la clase Actividad hay que tener en cuenta esta restricción, y escriba una instrucción para chequearla. Puede utilizar la clase Checkers.
4. Dados un objeto e de tipo Espacio y un objeto a de tipo Actividad, escriba las instrucciones que realizan lo que se pide en cada caso (si algo de lo que se pide no es posible, indíquelo):
 - a) Crear un espacio e con el nombre "A3.11" y la capacidad 175
 - b) Aumentar un 10% el número máximo de asistentes de la actividad a
 - c) Cambiar el lugar de la actividad a por el espacio e
 - d) Aumentar en 10 la capacidad del espacio e
 - e) Retrasar 30 minutos la hora de comienzo de la actividad a
 - f) Mostrar en pantalla la capacidad del espacio donde tiene lugar la actividad a

- Escriba en la clase Actividad un método **tipoAcceso** que devuelva una cadena de caracteres con el valor “libre” si la actividad es de tipo CONFERENCIA o TALLER, “restringido” si es de tipo REUNION, y “discrecional” si es de tipo OTRO. Use la estructura switch.
- El tipo Actividad tiene un orden natural dado por la fecha, y a igualdad de fecha por la hora de comienzo. Complete el código de la clase Actividad que se muestra para implementar este orden.

```
public class Actividad {
    ...
    public int compareTo( ) {
        ...
    }
}
```

- Escriba los siguientes métodos de la clase Actividad:
 - Un constructor que reciba parámetros para la hora de comienzo, el tipo de actividad y el lugar, y cree una actividad con la fecha de hoy, 0 asistentes y sin inscripción previa
 - El método consultor de la propiedad lugar
 - El método modificador de la propiedad lugar
 - El método consultor de la propiedad capacidadMaxima
- El tipo Actividad tiene un constructor a partir de String que crea un objeto a partir de una cadena con el formato “12/3/2024, 11:30, Otro, A3.11, 175, 70, false”. Complete las instrucciones del constructor a partir de String que se indican:

```
public Actividad(String s) {
    String[] sp = s.split(",");
    Checkers.check("Cadena con formato no válido", sp.length == 7);
    ...

    TipoActividad tipo =
    String nombre =
    Integer capacidad =
    Espacio e =
    Integer asistentes =

    ...
}
```

- El tipo ActividadDePago extiende al tipo Actividad, añadiendo una propiedad **precio** de tipo Double y otra propiedad **recaudación** de tipo Double, que se calcula multiplicando el precio por el número de asistentes.
 - Escriba un constructor de la clase ActividadDePago que reciba un parámetro por cada propiedad básica del tipo.
 - Escriba el método consultor de la propiedad derivada recaudación.
- Sea listaActividades una lista de actividades, conjuntoActividades un conjunto ordenado de actividades, y a una actividad. Indique para cada una de las siguientes instrucciones si es correcta o incorrecta:
 - listaActividades.add(0, a);
 - Actividad ultima = listaActividades.last();
 - listaActividades.reverse();
 - listaActividades.set(0, a);
 - Boolean vacio = conjuntoActividades.isEmpty();
 - Actividad primera = conjuntoActividades.get(0);
 - Collections.sort(actividades);
 - Integer n = Collections.size(listaActividades);



11. Escriba las instrucciones necesarias para realizar las siguientes operaciones:
- Cree una lista de actividades con el nombre `actividades`, copiando el contenido de la lista `l`
 - Añada la actividad `a` en la primera posición de la lista `actividades`
 - Obtenga en la variable `pos` la posición de la última aparición de la actividad `a` en la lista `actividades`
 - Elimine de la lista `actividades` las actividades que están presentes en el conjunto de actividades `otrasActividades`
 - Ordene la lista `actividades` según su orden natural
 - Elimine el primer elemento de la lista `actividades`
 - Elimine el último elemento de la lista `actividades`
 - Invierta las posiciones de los elementos de la lista `actividades`
12. Sea el siguiente código:
- ```
List<Integer> lista = List.of(5, 2, -3, 0, 9, -7, 8, 10, 1, 5, -4, 14, 3);
Integer res = null;
for (Integer n: lista) {
 if (res == null || n > res) {
 res = n;
 }
}
```
- ¿Cuál es el valor final de la variable `res`?
  - Utilice la clase `Collections` para realizar la misma operación
13. Sea `espacios` una lista de espacios. Escriba un bucle que obtenga la suma de las capacidades de los espacios cuyo nombre comience por la letra 'A'.
14. Escriba un método del tipo `Evento` que reciba como parámetro una fecha y devuelva una lista con los nombres de los espacios en los que se celebren actividades en esa fecha.
15. Escriba un método del tipo `Evento` que reciba como parámetro un tipo de actividad y devuelva la media de asistentes a las actividades de ese tipo que requieran inscripción previa. Si no hay actividades de ese tipo que requieran inscripción, el valor devuelto será null.
16. Escriba un método del tipo `Evento` que reciba como parámetro un numero entero `n` y devuelva un valor `true` si todas las actividades tienen un número de asistentes superior al  $n\%$  de la capacidad del espacio donde se realizan, o `false` en caso contrario.