



EXAMEN FINAL - PARTE 2

PROGRAMACIÓN BÁSICA

CONTEXTO

El teatro "ContemplArte" está buscando modernizar su sistema de boletería, para que después de cada función el sistema sea capaz de generar unas estadísticas tales como:

- Cantidad total de boletas vendidas
- Cantidad total de dinero recaudado en una función
- Cantidad total de boletas general vendidas
- Cantidad total de dinero recaudado por boletas general
- Cantidad total de boletas VIP vendidas
- Cantidad total de dinero recaudado por boletas VIP

Usted ha sido contratado como Desarrollador en Java, porque ha logrado demostrar habilidades de desarrollo en este lenguaje de programación y se le ha concedido implementar la clase correspondiente a boleta, y un método que permita conocer algunas estadísticas sobre una función.

El método que genera las estadísticas de una colección de instancias de la clase Boleta debe retornar en el siguiente orden en un array de enteros:

- Cantidad total de boletas vendidas
- Cantidad total de dinero recaudado en una función
- Cantidad total de boletas general vendidas
- Cantidad total de dinero recaudado por boletas general
- Cantidad total de boletas VIP vendidas
- Cantidad total de dinero recaudado por boletas VIP

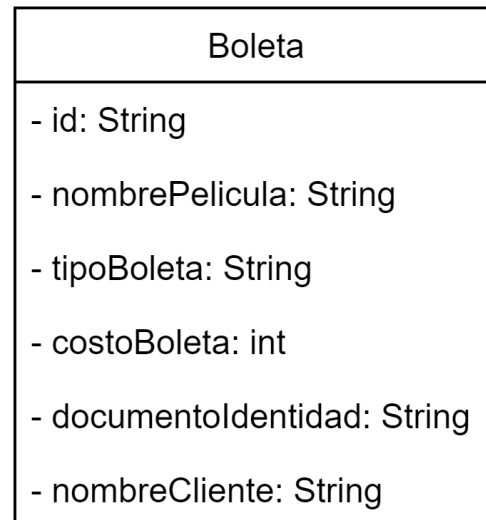
TAREAS

Realizar un método en Java (Llamado statisticsCinema) que le permitirá al administrador del teatro conocer unas estadísticas básicas sobre sus funciones, teniendo en cuenta las siguientes especificaciones:





Cada función se va a representar como un `ArrayList` de boletas, donde cada elemento es una instancia de la clase `Boleta` de la cual el equipo de ingeniería de software le hace entrega del siguiente diagrama de clases:



Recuerde que los métodos relacionados al constructor, getters y setters son obviados en el diagrama de clases, pero deberán ser incluidos en el código; estos métodos deben ser creados con el estándar camel case, por ejemplo, si el atributo se llama `tipoBoleta`, sus métodos correspondientes a get y set serían `getTipoBoleta` y `setTipoBoleta`.

Nota: Los parámetros del método constructor deberán aparecer en el mismo orden en que aparecen en el diagrama de clases, es decir, para el caso de `Enemigo` se espera que el constructor sea: `public Boleta(String id, String nombrePelicula, String tipoBoleta, int costoBoleta, String documentoIdentidad, String nombreCliente)...`

A partir de este `ArrayList`, usted deberá calcular:

- Cantidad total de boletas vendidas (dato de tipo `int`)
- Cantidad total de dinero recaudado en una función (dato de tipo `int`)
- Cantidad total de boletas general vendidas (dato de tipo `int`)
- Cantidad total de dinero recaudado por boletas general (dato de tipo `int`)
- Cantidad total de boletas VIP vendidas (dato de tipo `int`)
- Cantidad total de dinero recaudado por boletas VIP (dato de tipo `int`)





ENTRADAS

Su método recibirá como parámetro un `ArrayList` de instancias de la clase `Boleta` que representa una función. NO CREE LOS DATOS DE ENTRADA, DEBE USAR LOS QUE SE RECIBEN COMO PARÁMETRO.

Recuerde que el método se llama `statisticsCinema`

SALIDAS

Su método debe retornar un `Array` de `int` donde:

- En la primera posición se guardará la cantidad total de boletas vendidas (dato de tipo `int`)
- En la segunda posición se guardará la cantidad total de dinero recaudado en una función (dato de tipo `int`)
- En la tercera posición se guardará la cantidad total de boletas general vendidas (dato de tipo `int`)
- En la cuarta posición se guardará la cantidad total de dinero recaudado por boletas general (dato de tipo `int`)
- En la quinta posición se guardará la cantidad total de boletas VIP vendidas (dato de tipo `int`)
- En la quinta posición se guardará la cantidad total de dinero recaudado por boletas VIP (dato de tipo `int`)

EJEMPLO

Suponga que se le entrega el siguiente Array:

```
ArrayList<Boleta> t1 = new ArrayList<>();  
t1.add(new Boleta("000", "Cars 3", "VIP", 24000, "1063987245", "Mat Rivera"));  
t1.add(new Boleta("001", "Cars 3", "VIP", 24000, "1063741243", "Johan Z"));  
t1.add(new Boleta("002", "Cars 3", "general", 15000, "71588654", "Paula H"));  
t1.add(new Boleta("003", "Cars 3", "general", 15000, "98547632", "Mauro"));  
t1.add(new Boleta("004", "Cars 3", "general", 15000, "147896301", "David M"));
```

Se espera que su algoritmo retorne un `Array` de tipo `int` con los siguientes datos:

run:

[5, 93000, 3, 45000, 2, 48000]

BUILD SUCCESSFUL (total time: 0 seconds)





Explicación: Se vendieron 5 boletas, se recaudó 93000 en total, de tipo general se vendieron 3 boletas, con ellas se recaudó 45000, de tipo VIP se vendieron 2, y con ellas se recaudó 48000.





NOTA ACLARATORIA

Usted podrá desarrollar la prueba en Netbeans. Al final debe copiar y pegar el código en la herramienta VPL, pero **NO** deberá subir archivos, es decir:

Modo incorrecto:

NO SUBIR NINGÚN ARCHIVO

Descripción Entrega **Editar** Ver entrega

Entrega

Comentarios

Seleccione un archivo... Tamaño máximo para archivos nuevos: 5MB

solucion.py

Puede arrastrar y soltar archivos aquí para añadirlos

Enviar Cancelar

Modo correcto:

Descripción Lista de entregas Similaridad **Probar actividad** LUGAR CORRECTO

Entrega **Editar** Ver entrega Calificación Lista entregas previas

★ Solution.java

```
1 public class Solution{
2     //ESTA CLASE NO TIENE MAIN
3
4
5     public static double[] reporte(double[] listaNotas) {
6         //EN ESTE ESPACIO PONER SU LÓGICA
7
8
9
10    }
11 }
```

¡TRIPULANTE MUCHOS ÉXITOS EN EL DESARROLLO DEL RETO!





ACLARACIÓN DE PLAGIO

El objetivo es que los tripulantes cuenten con una oportunidad de aprendizaje relacionada con la programación. La colaboración académica es buena mientras no se lleve a un engaño académico, ya que el engaño académico inflige las buenas conductas del saber y del aprendizaje. El engaño académico hace referencia al plagio o envío de ideas que no son propias.

Colaborar implica compartir ideas, explicar a alguien cómo podría hacer su trabajo (más no hacer el trabajo por el otro) y ayudar al otro si tienes problemas a la hora de ejecutar o encontrar errores en el código.

En aras de evitar el plagio se recomienda colaborar pero no compartir su código o proyecto, no compartir sus soluciones, no usar un código encontrado en internet u otras fuentes que las propias. (Mason, Gavrilovska, y Joyner, 2019)

Los ejercicios enviados a verificación deben cumplir con la política antiplagio. Es decir, cualquier envío que sea una copia textual de otro trabajo puede ser suspendido o no aprobado por parte del equipo evaluador. El acto de copiar material de otro estudiante es un comportamiento inaceptable para el desarrollo de las competencias individuales y su progreso en este curso.

Referencia.

Mason, T., Gavrilovska, A., y Joyner, D.A. (2019). *Collaboration vesus cheating. 50th ACM Technical Symposium on Computer Science Education SIGCSE 2019*, Mineapolis, MN. DOI: 10.1145/3287324.3287443

