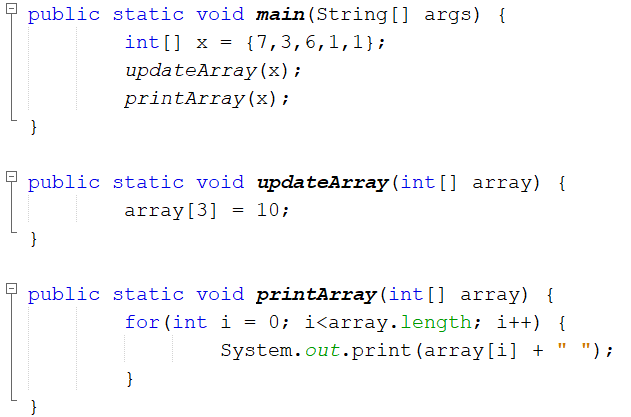
Instituto Tecnológico y de Estudios Superiores de Monterrey  
Informática II – Laboratorio Segundo Parcial

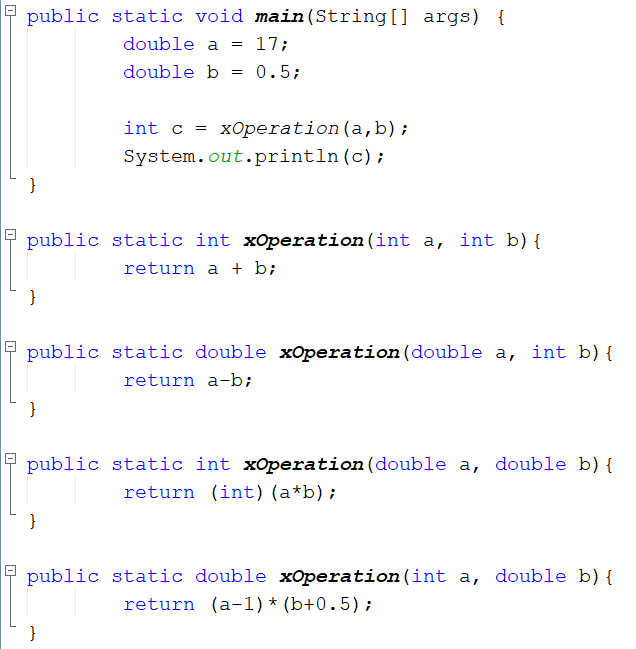
**Sección 1. Selecciona la mejor opción.**

1. **Indica el resultado de ejecutar el siguiente código.**

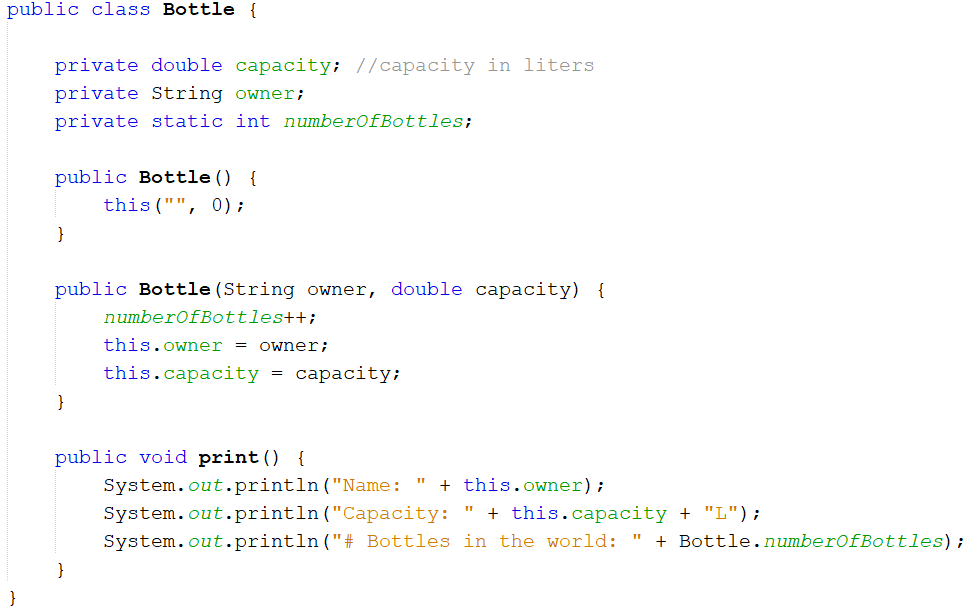


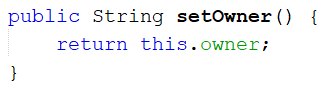
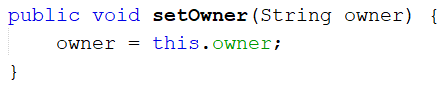
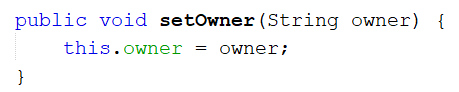
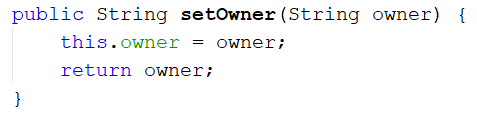
* 1. 7 3 6 1 1
  2. 7 3 10 1 1
  3. 7 3 6 10 1
  4. 7 3 10 1

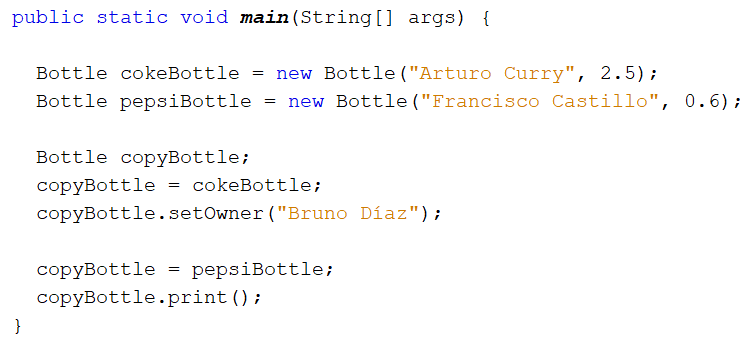
1. **Indica el resultado de ejecutar el siguiente código**



1. 17.0
2. 8
3. 16.0
4. 16
5. 17
6. 8.0

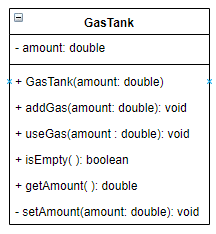


1. **Elige la opción que mejor representa un setter para la variable owner:**
   1. 
   2. 
   3. 
   4. 
2. **Tomando como resultado la clase Bottle, y tu elección del método setOwner, cuál sería el resultado de ejecutar el siguiente código:**



1. Name: Francisco Castillo  
   Capacity: 0.6L  
   # Bottles in the world: 2
2. Name: Arturo Curry  
   Capacity: 2.5L  
   # Bottles in the world: 1
3. Name: Francisco Castillo  
   Capacity: 0.6L  
   # Bottles in the world: 1
4. Name: Bruno Díaz  
   Capacity: 0.6L  
   # Bottles in the world: 1

**Sección 2:** Diseña una clase **GasTank** que sirva para representar un tanque de gasolina. Diseña la clase de acuerdo con el siguiente diagrama UML:



* **Constructor:** Este método debe recibir como parámetro de entrada una cantidad **amount**, y llame al método **setAmount** para actualizar la variable de instancia **amount.**
* **addGas**:deberá incrementar la cantidad de gasolina en el tanque en la cantidad recibida como parámetro**.** Asegúrate de validar que sólo se procesen valores positivos.
* **useGas:** deberá reducir la cantidad de gasolina en el tanque en la cantidad recibida como parámetro. Asegúrate de que sólo se procesen valores positivos.
* **isEmpty:** Deberá devolver **true** cuando la cantidad de gasolina en el tanque sea menor a 0.1. De lo contrario, deberá retornar **false.**
* **getAmount:** Getter para la variable amount.
* **setAmount:** Método privado (sólo será usado por el constructor) que actualice la variable **amount** siempre y cuando el parámetro recibido sea mayor o igual a 0.

**Sección 3. Resuelve el siguiente caso.**

Cineplus, una nueva cadena de complejos de cine,ha decidido abrir su primera sucursal en la ciudad de Monterrey. Su concepto único está basado en ofrecer boletos de cine a un costo variable dependiendo de la disponibilidad de asientos en cada función.

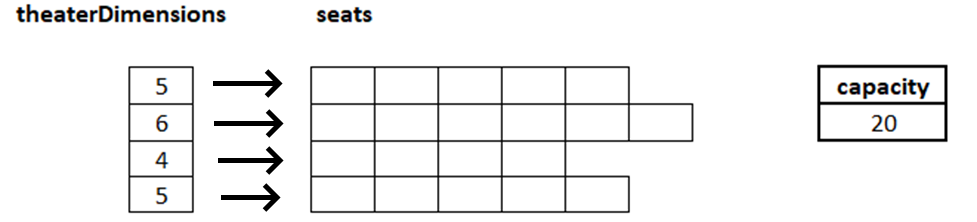
Diseña y codifica una clase llamada CinemaShow que permita modelar una función de cine en un complejo de Cineplus. La clase deberá contar con los atributos definidos a continuación. Elige los tipos de datos que mejor se adecúan para representar la siguiente información. Utiliza las mejores prácticas y conceptos de la programación orientada a objetos. Considera si cada método debe ser estático o de instancia.

* Variable movieName que sirva para almacenar el nombre de la película que se va a proyectar.
* Variable movieDate que sirva para representar la fecha en la que se proyectará la película.
* Variable movieTimeque sirva para representar la hora en la que se proyectará la película.
* Variable capacity que permita almacenar el aforo de la sala de cine (cantidad de personas que caben en la sala).
* Variable soldTickets que mantenga un registro actualizado de la cantidad de asientos vendidos de la función al momento.
* Arreglo bidimensional seatsque sirva para llevar el control de los asientos vendidos.

Adicionalmente, la clase deberá contener las siguientes acciones:

* Método constructor que reciba e inicialice las variables: movieName**,** movieDateymovieTime**.**
* Métodos setter diferentes para las variables: movieName**,** movieDateymovieTime.
* Métodos getter diferentes para las variables: capacityysoldTickets**.**
* Método initializeSeatsque reciba como parámetro de entrada un arreglo de enteros theaterDimensions, y que inicialice el arreglo seats**.** Adicionalmente, deberá calcular el aforo de la sala y almacenarlo en la variable de instancia capacity**.**

El arreglo recibido theaterDimensionsrepresentará la cantidad de asientos en cada fila de la sala. Por ejemplo:



* Método double assignPrice(int soldTickets, int totalTickets) que calcule y retorne el precio de un boleto de acuerdo con la siguiente fórmula:

Por ejemplo, el boleto #36 se vendería en:

* Método boolean sellSeat(int row, int column)que reciba como parámetro de entrada dos enteros: row y column, calcule el precio del boleto (ver método assignPrice), y lo almacene en la fila y columna recibida. Posteriormente deberá actualizar la variable soldTickets**.**
* Sobrecarga el método boolean sellSeat() para que asigne automáticamente un espacio vacío en la sala.

**El arreglo seats no debe ser retornado directamente por un método getter. ¿Cuál es el riesgo de implementar un getter para dicha variable y cómo pudiera mitigarse? Justifica tu respuesta.**

|  |
| --- |
|  |