Universidad del Valle de Guatemala

Programación Orientada a Objetos sección 40

Mar Isabel Orozco Moscoso 24967

**Ejercicio 1: La Academia**

**Análisis y diseño**

1. **¿Qué debe hacer su sistema? Haga una lista con los requisitos funcionales.**

* Configuración inicial: nombre del venue, capacidad de cada localidad, fecha del concierto.
* Vender boleto:
  + El comprador ingresa su nombre, email, cantidad de boletos a comprar (máximo 6) y presupuesto máximo.
  + El comprador debe seleccionar la localidad del ticket (balcón 1, platea o balcón VIP)
  + Verificar si el boleto puede venderse:
    - Validar espacio en la localidad: si no hay espacio no se puede vender nada, si no hay suficiente espacio se vende la cantidad máxima de boletos que se pueda.
    - Validar si el presupuesto ingresado alcanza para comprar los boletos deseados (si no, se termina el proceso).
      * Balcón 2: Q300
      * Platea: Q600
      * Balcón VIP: Q1800
  + Construir el número del ticket para cada boleto usando para esto la fecha (ddmmAAAA) del día que se compraron las entradas más el correlativo (número de compra ese día).
  + Confirmar compra de boleto
* Consultar disponibilidad total y precios: se muestran cuántos boletos se han vendido en cada localidad y cuantos boletos hay disponibles para las 3 localidades.
* Consultar disponibilidad de una localidad específica y precio
* Reporte de caja: Mostrar cuánto dinero se ha generado.
* Salir.

1. **¿Qué clases necesita para poder cumplir con esta lista de requisitos? ¿Qué métodos y atributos tienen cada clase?**

* **Venue**

**Atributos:**

* + Nombre del venue (String)
  + Balcón (Localidad)
  + Platea (Localidad)
  + VIP (Localidad)
  + Fecha Concierto (date)
  + Ventas hoy (int)

**Métodos:**

* + VerificarVenta(comprador Comprador): valida espacio, valida presupuesto, devuelve un boolean
  + validarEspacio(comprador Comprador): compara la disponibilidad con los boletos solicitados para la localidad indicada, devuelve falso si la disponibilidad es menor y verdadero si es mayor.
  + validarPresupuesto(comprador Comprador): multiplica el precio de la localidad por la cantidad de boletos y lo compara con el presupuesto. Devuelve un boolean.
  + Venta(comprador Comprador): crea los boletos pedidos, actualiza las ventas de hoy, añade las ventas a la localidad correspondiente y actualiza el atributo de venue, devuelve la lista de boletos.
  + Confirmación(lista de boletos): devuelve un string concatenando los toString de los 6 boletos.
  + Consultar\_disponibilidad(): devuelve el toString de cada localidad
  + Consultar\_disponibilidad\_específica(localidad Localidad): devuelve el toString de la localidad
  + Reporte\_caja: devuelve un double (cuánto dinero se ha generado).
* **Localidad**

**Atributos:**

* + Nombre de la localidad (String)
  + Capacidad (Int)
  + Vendidos (Int)
  + Precio (double)

**Métodos**

* toString: devuelve un string con la disponibilidad y el precio
* disponibilidad: devuelve el int capacidad - vendidos
* **Comprador**

**Atributos:**

* + Nombre (String)
  + Email (String)
  + Cantidad boletos (int)
  + Presupuesto máximo (double)
  + Localidad (Localidad)
* **Boleto**

**Atributos:**

* + Localidad (Localidad)
  + Número Asiento (String)
  + Comprador (Comprador)
  + Venue (Venue)

**Métodos**

* + numeroTicket()genera el número de ticket con la fecha de hoy, y las ventas de hoy.
  + toString()
* **Main**