

Tema 1 - Exercici Avaluable 1

Volem implementar un conjunt de classes que permeti gestionar un sistema de venda d'entrades pels diferents espectacles que es poden representar en un teatre. Suposarem que cada espectacle es pot representar en diferents dies, amb un preu que podria ser diferent per cada representació. El teatre té un nº total de seients que és el mateix per totes les representacions. Quan un espectador compra entrades per una representació d'un espectacle se li assignaran de forma automàtica els seients lliures necessaris en funció del nº d'entrades, seguint el procediment que s'explica a continuació. Cada seient ve identificat per un nº de seient entre 1 i el nº total de seients del teatre.

Per assignar els seients a un espectador es seguirà aquest procediment:

- Primer, buscarem si hi ha suficients seients lliures amb numeració consecutiva per cobrir el nº total d'entrades que es volen comprar. Si és el cas, s'assignaran tots els seients adjacents a partir del primer seient on es compleixi aquesta condició.
- Si no es poden trobar suficients seients adjacents, s'assignaran els primers seients que estiguin lliures (començant pel seient nº 1) fins a cobrir el total d'entrades, encara que no corresponguin a seients adjacents. En aquest cas, s'aplicarà un descompte del 10% en el preu final de les entrades.

Si un espectador compra més de 5 entrades se li aplica un descompte del 5%, que es pot acumular al descompte per assignar seients no adjacents

Tema 1 - Exercici Avaluable 1

Tindrem una classe que s'anomenarà `VendaEntrades` per gestionar la venda de les entrades pels espectacles. Aquesta classe haurà de tenir aquesta interfície pública:

- Un constructor que permeti inicialitzar el sistema de venda d'entrades, agafant com a paràmetre el nº total de seients del teatre.
- Un mètode per afegir un espectacle a la llista de representacions del teatre, amb aquesta capçalera:

```
void afegeixEspectacle(const string& nom, const string& dia, float preu);
```

- Un mètode per gestionar la compra d'entrades per un espectacle amb aquesta capçalera:

```
EstatCompra compraEntrades(const string& nom, string& dia, int nEntrades, float& preu,  
                             int seients[MAX_SEIENTS_COMPRA]);
```

Ha de gestionar la compra del nº d'entrades que s'indiquen per un determinat espectacle identificat per dia i hora. La selecció dels seients s'ha de fer segons el protocol que s'ha definit abans. Retorna si la compra s'ha pogut fer correctament o no (indicant el motiu), el preu total de la compra segons les regles que s'han indicat abans, i quins seients s'han assignat dins de l'array `seients`.

El valor de retorn s'indica amb el tipus enumerat `EstatCompra` que es declara més endavant. El valor `COMPRA_OK` indica que la compra s'ha pogut realitzar correctament, el valor `ERROR_ESPECTACLE` indica que la compra no s'ha pogut fer perquè l'espectacle no existeix i el valor `ERROR_SEIENTS` indica que la compra no s'ha pogut fer perquè no queden suficients seients lliures.

Tema 1 - Exercici Avaluable 1

- Un mètode per cancel·lar una compra d'entrades feta prèviament amb aquesta capçalera:

```
EstatCompra anulaCompra(const string& nom, const string& dia, int nEntrades,  
                        int seients[MAX_SEIENTS_COMPRA]);
```

Ha d'anul·lar una compra feta prèviament per un espectacle identificat per dia i hora. Els seients que s'han d'anul·lar s'indiquen a l'array `seients` i `nEntrades` és el nº de seients que s'havien comprat.

El valor de retorn s'indica amb el tipus enumerat `EstatCompra`. El valor `COMPRA_OK` indica que l'anul·lació s'ha pogut realitzar correctament, el valor `ERROR_ESPECTACLE` indica que l'anul·lació no s'ha pogut fer perquè l'espectacle no existeix i el valor `ERROR_SEIENTS` indica que l'anul·lació no s'ha pogut fer perquè alguns dels seients especificats dins de l'array `seients` no estan reservats.

- Un mètode per recuperar l'ocupació de tots els seients per un espectacle determinat amb aquesta capçalera:

```
void recuperaOcupacio(const string& nom, const string& dia,  
                    bool ocupacioSeients[MAX_SEIENTS]);
```

Retorna a l'array `ocupacioSeients`, l'estat d'ocupació de tots els seients per l'espectacle que s'indica. `True`indrà dir que el seient està ocupat i `false` que està lliure.

```
typedef enum  
{  
    COMPRA_OK,  
    ERROR_ESPECTACLE,  
    ERROR_SEIENTS  
} EstatCompra;
```

Tema 1 - Exercici Avaluable 1

1. Declareu una classe anomenada `Espectacle` que permeti guardar i gestionar tota la informació de cadascun dels espectacles que es representen al teatre. Penseu quins atributs heu de posar a la part privada i definiu la interfície pública de la classe que considereu necessària.
2. Utilitzant la classe `Espectacle`, declareu la part privada de la classe `VendaEntrades` perquè puguem gestionar la venda d'entrades per qualsevol espectacle.
3. Implementeu tots els mètodes de les classes `VendaEntrades` i `Espectacle`.