

# Tema 1 - Exercici Avaluable 3

Volem implementar un conjunt de classes per poder **gestionar l'ocupació de les aules de l'escola** i poder saber si estan reservades o no per un dia i hora concrets.

Les **reserves de les aules** es poden fer per qualsevol dia de la setmana, de dilluns a divendres, en horari des de les 8 del matí fins a les 9 de la nit. Les reserves sempre es faran per hores completes començant i acabant a les hores en punt, però una reserva pot ser per més d'una hora consecutiva (per exemple des de les 9 del matí fins a les 12). Per poder saber per quines setmanes es fa la reserva, cada setmana del curs s'identifica per un nº que indica el nº de setmana des de l'inici del curs acadèmic. Quan es fa una reserva s'indica el nº de setmana d'inici i de final de la reserva (per exemple fer una reserva pels dimecres des de les 10 fins a les 12, des de la setmana nº 1 fins a la setmana nº 10).

De moment, suposem que **tenim creada la classe Reserva** que us mostrem a continuació per guardar la informació d'una reserva per una aula:

```
class Reserva
{
private:
    string m_diaSetmana;
    int m_horaInici;
    int m_horaFinal;
    int m_nSetmanaInici;
    int m_nSetmanaFinal;
};
```

# Tema 1 - Exercici Avaluable 3

Utilitzant aquesta classe **volem declarar i implementar una classe Aula** que ens permeti gestionar la informació d'una aula i de les reserves que s'hi han fet. Aquesta classe ha de poder guardar el codi de l'aula, la seva capacitat i la informació de totes les reserves que s'han fet. Podeu suposar que una aula no tindrà més de 500 reserves actives. Aquesta classe ha de tenir mètodes que permetin fer les següents accions:

- **Afegir una reserva** per un dia, hores i setmanes concrets. La reserva només es podrà fer si la franja horària que es vol reservar no se solapa amb cap de les reserves que ja estan fetes per aquesta aula. Per exemple, si tenim ja feta una reserva pels dimecres de 10 a 12 per les setmanes des de la nº 10 a la nº 15, aquesta reserva se solaparia i seria incompatible amb una nova reserva que es vulgui fer pels dimecres de les setmanes nº 1 fins la nº 12 de 11 a 13. Si la reserva no es pot fer s'ha d'indicar d'alguna forma amb el valor de retorn.
- **Anul·lar una reserva** ja feta per aquesta aula, a partir de la informació de la reserva (dia, hora d'inici i final i setmana d'inici i final). Si la reserva no existeix s'ha d'indicar d'alguna forma amb el valor de retorn.

També hem de guardar la informació de les **assignatures** que es fan a l'Escola per poder fer l'assignació d'aules a les assignatures. Les assignatures estaran identificades per un codi i per poder fer l'assignació de les aules necessitarem saber el número d'estudiants matriculats i el seu horari (dia de la setmana, hora d'inici i final) i el període en què s'imparteix (setmana inicial i final). Per simplificar, suposarem que cada assignatura té un sol grup de classe i que es fa en un horari únic (totes les classes concentrades en un sol dia de la setmana i una única franja horària).

# Tema 1 - Exercici Avaluable 3

Per gestionar tot el procés de reserva d'aules i assignació d'aules a assignatures, volem declarar una classe **GestioAules**, que agrupi la informació de totes les aules i de totes les assignatures i tingui la **interfície pública** següent:

- Un mètode per inicialitzar la informació de totes les aules a partir de la informació guardada en un fitxer:

```
void inicialitzaAules(const string& nomFitxer);
```

El fitxer guardarà la informació de les aules i de les reserves fetes fins aquest moment seguint aquest format:

```
codi_aula capacitat_aula n_reserves
dia_setmana_1 hora_inici_1 hora_final_1 setmana_inici_1 setmana_final_1
dia_setmana_2 hora_inici_2 hora_final_2 setmana_inici_2 setmana_final_2
...
dia_setmana_N hora_inici_N hora_final_N setmana_inici_N setmana_final_N
codi_aula capacitat_aula n_reserves
dia_setmana_1 hora_inici_1 hora_final_1 setmana_inici_1 setmana_final_1
...
dia_setmana_N hora_inici_N hora_final_N setmana_inici_N setmana_final_N
...
```

Per cada aula, la primera línia conté el codi i la capacitat de l'aula i el nº de reserves fetes fins ara per aquella aula. Per cada reserva hi ha una línia amb el fitxer amb tota la informació de la reserva (dia, hores i setmanes).

## Tema 1 - Exercici Avaluable 3

- Un mètode per inicialitzar la informació de totes les assignatures a partir de la informació guardada en un fitxer:

```
void inicialitzaAssignatures(const string& nomFitxer);
```

El fitxer guardarà la informació de les aules i de les reserves fetes fins aquest moment seguint aquest format:

```
codi_assignatura_1 n_matriculats dia_setmana hora_inici hora_final setmana_inici setmana_final  
codi_assignatura_2 n_matriculats dia_setmana hora_inici hora_final setmana_inici setmana_final  
...  
codi_assignatura_n n_matriculats dia_setmana hora_inici hora_final setmana_inici setmana_final
```

A cada línia del fitxer hi ha la informació completa d'una assignatura: codi, número de matriculats i horari.

- Un mètode per assignar una aula a una assignatura:

```
string assignaAula(const string& codiAssignatura);
```

S'ha d'assignar l'aula més petita entre totes les que tenen capacitat suficient perquè hi càpiguen el número d'alumnes matriculats a l'assignatura i estan lliures a l'horari de l'assignatura. Si hi ha alguna aula disponible s'ha d'afegir la reserva a la llista de reserves de l'aula i s'ha de retornar el codi de l'aula reservada. Si no es pot reservar cap aula, ha de retornar un string buit.

- Un mètode per anul·lar l'assignació d'una aula a una assignatura:

```
bool eliminaReserva(const string& codiAssignatura);
```

S'ha d'eliminar la reserva de l'aula assignada per l'assignatura que es passa com a paràmetre. Si l'assignatura no té aula assignada s'ha de retornar false, i true en cas contrari.

# Tema 1 - Exercici Avaluable 3

- Un mètode per recuperar l'aula assignada a una determinada assignatura:

```
string aulaAssignatura(const string& codiAssignatura);
```

Si no hi ha aula assignada a l'assignatura s'ha de retornar un string buit.

- Un mètode per recuperar totes les assignatures que es fan a una aula determinada:

```
void assignaturesAula(const string& codiAula, int& nAssignatures, string assignatures[]);
```

S'ha de retornar el nom de totes les assignatures dins de l'array assignatures i el número d'assignatures a nAssignatures.

- Un mètode per escriure a un fitxer la informació de totes les reserves fetes per una aula:

```
void reservesAula(const string& codiAula, const string& nomFitxer);
```

El fitxer guardarà la informació de les reserves seguint aquest format:

```
dia_setmana_1 hora_inici_1 hora_final_1 setmana_inici_1 setmana_final_1
```

```
dia_setmana_2 hora_inici_2 hora_final_2 setmana_inici_2 setmana_final_2
```

```
...
```

```
dia_setmana_N hora_inici_N hora_final_N setmana_inici_N setmana_final_N
```

# Tema 1 - Exercici Avaluable 3

1. Feu la **declaració de la classe Aula** indicant els atributs i els mètodes que hauria de tenir la classe. Al fer la declaració de la classe tingueu en compte d'aplicar tots els conceptes que hem anat explicant:
  - Separació privat/públic.
  - Declarar i implementar un constructor amb paràmetres per inicialitzar les dades bàsiques de l'aula.
  - Incloure els getters/setters necessaris.
  - Declarar els paràmetres i mètodes constants que faci falta.
  - Pensar i especificar bé els paràmetres i valor de retorn dels mètodes que faci falta declarar.
2. Afegiu a la **classe Reserva** els mètodes públics que penseu que seran necessaris per poder implementar els mètodes de la classe Aula. Tingueu en compte els mateixos punts que hem comentat per la classe Aula. A més a més:
  - Declareu un operador per poder llegir les dades de la reserva d'un fitxer (per poder-lo utilitzar quan inicialitzem les reserves des de fitxer).
  - Declareu un operador per poder comparar si dues reserves són iguals (per poder-lo utilitzar quan anul·lem una reserva).
3. Feu la **declaració de la classe Assignatura** (mètodes i atributs) tenint en compte els mateixos aspectes que a les classes anteriors. Penseu quina interfície pública necessiteu per implementar els mètodes de GestioAules. Podeu utilitzar la classe Reserva per guardar l'horari de l'assignatura.
4. Feu la **declaració de la classe GestioAules**, amb els mètodes de la interfície pública que us hem donat abans i tots els atributs que siguin necessaris.
5. Feu la **implementació** de tots els **mètodes** de la interfície pública de la **classe GestioAules**, i de tots els mètodes de les **altres classes** que siguin també necessaris.