

## Tema 2 – Exercici recuperació 2

Volem crear un conjunt de classes per poder gestionar les reserves d'habitació que es fan a un hotel.

1. Declareu i implementeu una classe Reserva per poder guardar les dades d'una reserva d'habitació. Aquesta classe ha de tenir atributs per guardar el nom del client que fa la reserva, les dates d'inici i de final de la reserva, el nº d'habitacions que es reserven i el preu de la reserva. A més a més haurà de tenir els constructors i tots els mètodes per modificar i recuperar els valors dels atributs que siguin necessaris per implementar la classe ReservesHotel que es descriu a continuació.

Per guardar els valors de les dates d'inici i final podeu fer servir la classe Data que us mostrem a continuació.

```
class Data
{
public:
    Data();
    Data(int dia, int mes, int any);
    Data(const Data &data);
    void setDia(int dia);
    void setMes(int mes);
    void setAny(int any);
    int getDia() const;
    int getMes() const;
    int getAny() const;
    bool dataValida() const;
    Data operator+(int nDies) const;
    bool operator==(const Data& data) const;
    bool operator<(const Data& data) const;
    Data &operator=(const Data& data);
};
```

## Tema 2 – Exercici recuperació 2

2. Utilitzant la classe Reserva de l'exercici anterior declareu i implementeu una classe ReservesHotel que permeti guardar totes les reserves que es fan en un hotel. A la classe hi hem de poder guardar el nom de l'hotel, el preu per dia de l'habitació, el nº d'habitacions de l'hotel i les dades de totes les reserves utilitzant la classe Reserva de l'exercici anterior.
- Feu la declaració de la classe amb tots els atributs que calgui per guardar les dades de l'hotel i les seves reserves. Utilitzeu un array dinàmic per guardar les dades de les reserves.
  - Implementeu un constructor que rebi com a paràmetres el nom de l'hotel, el preu per dia de l'habitació, el nº d'habitacions de l'hotel i el nº màxim de reserves que hem de poder guardar a l'array dinàmic de reserves.
  - Implementeu un constructor de còpia.
  - Implementeu l'operador d'assignació.
  - Implementeu el destructor, que alliberi correctament la memòria dinàmica utilitzada.

## Tema 2 – Exercici recuperació 2

- Implementeu un mètode `llegeixReserves` que permeti llegir d'un fitxer totes les dades de les reserves de l'hotel. Ha de rebre com a paràmetre el nom del fitxer. Si ja hi havia reserves a les dades de l'hotel, les reserves que es llegeixin del fitxer s'afegiran a les que ja hi havia. Si es supera el límit de reserves màxim per hotel que es pot guardar a l'array dinàmic, es para de llegir i s'ignoren les reserves que quedin al fitxer.

El fitxer té el següent format (les dates estan guardades al fitxer en format DD/MM/YYYY):

NOM\_CLIENT\_1

DATA\_ENTRADA\_1

N\_HABITACIONS\_1

N\_DIES\_1

NOM\_CLIENT\_2

DATA\_ENTRADA\_2

N\_HABITACIONS\_2

N\_DIES\_2

...

NOM\_CLIENT\_N

DATA\_ENTRADA\_N

N\_HABITACIONS\_N

N\_DIES\_N

## Tema 2 – Exercici recuperació 2

- Implementeu un mètode `nReservesDia` que retorni quantes habitacions hi ha reservades per un dia concret que es passa com a paràmetre utilitzant un objecte de la classe `Data`.
- Implementeu un mètode `afegeixReserva` que permeti afegir una nova reserva a la llista de reserves. Aquest mètode rebrà com a paràmetres el nom del client, la data d'entrada, el nº de dies d'estada i el nº d'habitacions a reservar. S'haurà de comprovar que a l'hotel quedin suficients habitacions lliures per tots els dies de l'estada. Per fer-ho podeu utilitzar el mètode `nReservesDia` de l'apartat anterior. Si la reserva és possible el mètode retornarà `true` i si no és possible retornarà `false`.
- Implementeu un mètode `consultaReserva` que rebi com a paràmetres el nom d'un client i una data d'entrada (utilitzant un objecte de la classe `Data`) i retorni, utilitzant paràmetres per referència, la resta de dades de la reserva que ha fet aquest client: data de sortida, nº d'habitacions i preu de la reserva. Si la reserva no existeix ha de retornar `false`, i si existeix, `true`.