Sessió 12

Tema 2

Exercici 2.9:

};

#include "Carta.h" 2.9 Donades les classes Carta i Joc: #include <stack> const int CORS = 0; #include <cstddef> const int DIAMANTS = 1; #include <iostream> const int PIQUES = 2; using namespace std; const int TREVOLS = 3; const int PALCARTABUIDA = 5; const int N CARTES = 13; class Carta const int N PALS = 4; {public: class Joc Carta(); { public: Carta(int pal, int valor); Joc(); Carta(const Carta& c); ~Joc(); ~Carta(); bool treureCartaBaralla(); bool esCartaBuida(); bool posarDestapadesABaralla(); void setCarta(int noupal, int nouvalor); bool posarCartaAPal(int pal); int getValor() const; void escriuJoc(); int getPal() const; void donaJoc(Carta& cBaralla, void escriuCarta() const; Carta& cDestapada, Carta& operator=(const Carta& c); Carta cPila[N PALS])const; bool operator==(const Carta& c); private: private: //m baralla; int m valor; //m destapades; int m pal;

};

//m pilaResultat[N PALS];

Exercici 2.9:

- 2.9 Donades les classes Carta i Joc: volem definir el joc del solitari de manera simplificada. Tenim:
- 4 pals: CORS=0; DIAMANTS = 1; PIQUES = 2; TREVOLS = 3;
- Cada pal te 13 valors: de l'1 al 13, essent el 11, 12 i 13 els corresponents a J, Q, K respectivament i l'1 es l'As.
- 4 piles, una per cada pal de cartes, a on anirem col·locant les cartes del pal que tenen per nom de forma ordenada, començant per l'as i acabant pel rei.
 Anomenades:
 - m_pilaResultat[CORS]
 - m pilaResultat[DIAMANTS]
 - m_pilaResultat[PIQUES]
 - m_pilaResultat[TREVOLS].
- **1 pila** de cartes que anomenarem **m_baralla**, on tindrem inicialment totes les cartes ordenades:

```
(Pal, Valor): (0,1),(1,1),(2,1),(3,1),(0,2),(1,2),(2,2),(3,2),...,(0,13),(1,13),(2,13),(3,13)
```

• **1 pila** de cartes que anomenarem **m_destapades**, on anirem posant les cartes de m_Baralla a mesura que les anem destapant. Quedant sempre a dalt la ultima carta destapada.

Exercici 2.9.1: Definició piles stl

```
#include "Carta.h"
2.9.1 Definiu les piles: m baralla,
                                             #include <stack>
   m destapades, m pilaResultat[N PALS]
                                             const int CORS = 0;
   de la classe joc com a piles de la llibreria
                                             const int DIAMANTS = 1;
   stl
                                             const int PIQUES = 2;
                                             const int TREVOLS = 3;
const int PALCARTABUIDA = 5;
                                             const int N CARTES = 13;
class Carta
                                             const int N PALS = 4;
{public:
                                             class Joc
  Carta();
                                             { public:
  Carta(int pal, int valor);
                                                 Joc();
  Carta(const Carta& c);
                                                 ~Joc();
  ~Carta();
                                                 bool treureCartaBaralla();
  bool esCartaBuida();
                                                 bool posarDestapadesABaralla();
  void setCarta(int noupal, int nouvalor);
                                                 bool posarCartaAPal(int pal);
  int getValor() const;
                                                 void escriuJoc();
  int getPal() const;
                                                 void donaJoc(Carta& cBaralla,
  void escriuCarta() const;
                                                               Carta& cDestapada,
  Carta& operator=(const Carta& c);
                                                               Carta cPila[N PALS])const;
  bool operator==(const Carta& c);
                                                 private:
 private:
                                                   //m_baralla;
  int m_valor;
                                                  //m destapades;
  int m pal;
                                                   //m pilaResultat[N PALS];
};
                                             };
```

Exercici 2.9.1: Solució definició piles stl

2.9.1 Definiu les piles: m_baralla, m_destapades, m_pilaResultat[N_PALS] de la classe joc com a piles de la llibreria stl

```
#include "Carta.h"
#include <stack>
                                      private:
const int CORS = 0;
                                        std::stack<Carta> m baralla;
                                        std::stack<Carta> m destapades;
const int DIAMANTS = 1;
                                        std::stack<Carta> m pilaResultat[N PALS];
const int PIQUES = 2;
const int TREVOLS = 3;
                                      };
const int N CARTES = 13;
const int N PALS = 4;
class Joc
{ public:
    Joc();
    ~Joc();
    bool treureCartaBaralla();
    bool posarDestapadesABaralla();
    bool posarCartaAPal(int pal);
    void escriuJoc();
    void donaJoc(Carta& cBaralla,
                 Carta& cDestapada,
                 Carta cPila[N PALS])const;
```

Exercici 2.9.2: Constructor per defecte

2.9.2 Implementar el constructor per defecte de Joc que inicialitzi la pila m_baralla tal com hem descrit: Joc();

```
(Pal, Valor): (0,1),(1,1),(2,1),(3,1),(0,2),(1,2),(2,2),(3,2),...,(0,13),(1,13),(2,13),(3,13)
           [top de la pila]
                                                               [bottom de la pila]
const int PALCARTABUIDA = 5;
                                                 #include "Carta.h"
class Carta
                                                 #include <stack>
{public:
                                                 const int CORS = 0;
 Carta();
                                                 const int DIAMANTS = 1;
 Carta(int pal, int valor);
                                                 const int PIQUES = 2;
 Carta(const Carta& c);
                                                 const int TREVOLS = 3;
  ~Carta();
                                                 const int N CARTES = 13;
  bool esCartaBuida();
                                                 const int N PALS = 4;
  void setCarta(int noupal, int nouvalor);
                                                 class Joc
  int getValor() const;
                                                 { public:
  int getPal() const;
                                                     Joc(); ...
  void escriuCarta() const;
                                                   private:
  Carta& operator=(const Carta& c);
                                                      //m baralla;
  bool operator==(const Carta& c);
                                                      //m destapades;
 private:
                                                      //m_pilaResultat[N_PALS];
  int m_valor;
                                                 };
  int m pal;
};
```

Exercici 2.9.2: Solució constructor per defecte

2.9.2 Implementar el constructor per defecte de Joc que inicialitzi la pila m_baralla tal com hem descrit: Joc();

```
Joc::Joc()
{
    for (int valor = N_CARTES; valor > 0; valor--)
        for (int pal = N_PALS - 1; pal >= 0; pal--)
        {
            Carta c;
            c.setCarta (pal, valor);
            m_baralla.push(c);
        }
}
```

Exercici 2.9.3: treureCartaBaralla()

2.9.3 Escriu el mètode de "Joc":bool treureCartaBaralla();

a)Si m_Baralla té cartes treu carta de m_Baralla i la posa a m_destapades, retornant true.

b)Si m_Baralla no té cartes ens retorna false.

```
const int PALCARTABUIDA = 5;
class Carta
{public:
 Carta();
 Carta(int pal, int valor);
 Carta(const Carta& c);
 ~Carta();
 bool esCartaBuida();
 void setCarta(int noupal, int nouvalor);
 int getValor() const;
 int getPal() const;
 void escriuCarta() const;
 Carta& operator=(const Carta& c);
 bool operator==(const Carta& c);
 private:
 int m valor;
 int m pal;
};
```

```
#include "Carta.h"
#include <stack>
const int CORS = 0;
const int DIAMANTS = 1;
const int PIQUES = 2;
const int TREVOLS = 3;
const int N CARTES = 13;
const int N PALS = 4;
class Joc
{ public:
    Joc(); ...
    bool treureCartaBaralla();
  private:
     //m baralla;
     //m_destapades;
     //m_pilaResultat[N_PALS];
};
```

Exercici 2.9.3: Solució treureCartaBaralla()

```
2.9.3 Escriu el mètode de "Joc":bool treureCartaBaralla();
a)Si m Baralla té cartes treu carta de m Baralla i la posa a m destapades, retornant
true.
b)Si m_Baralla no té cartes ens retorna false.
// Treu una carta de la baralla i la posa a m_destapades
bool Joc::treureCartaBaralla()
     bool movimentFet = false;
     if (!m_baralla.empty())
         m_destapades.push(m_baralla.top());
         m_baralla.pop();
         movimentFet = true;
     return movimentFet;
```

Exercici 2.9.4: posarCartaAPal()

2.9.4 Escriu un mètode de "Joc" que passa una carta de m_destapades a m_pilaResultat[pal], si és possible. Per fer-ho primer comprova que m_destapades tingui cartes i, si les té, que la primera sigui del pal correcte i el número sigui correlatiu al que ja està a m_pilaResultat[pal], o un as en cas de no haver-hi cap carta a m_pilaResultat[pal]. Si el moviment es pot fer retorna true i false en cas contrari.

```
const int PALCARTABUIDA = 5;
                                                 #include "Carta.h"
class Carta
                                                 #include <stack>
{public:
                                                  const int CORS = 0;
 Carta();
                                                  const int DIAMANTS = 1;
 Carta(int pal, int valor);
                                                 const int PIQUES = 2;
                                                 const int TREVOLS = 3;
 Carta(const Carta& c);
 ~Carta();
                                                 const int N CARTES = 13;
  bool esCartaBuida();
                                                  const int N PALS = 4;
  void setCarta(int noupal, int nouvalor);
                                                  class Joc
  int getValor() const;
                                                  { public:
  int getPal() const;
                                                      Joc(); ...
                                                      bool posarCartaAPal(int pal);
  void escriuCarta() const;
 Carta& operator=(const Carta& c);
                                                    private:
  bool operator==(const Carta& c);
                                                      //m_baralla;
 private:
                                                      //m destapades;
  int m valor;
                                                       //m pilaResultat[N PALS];
  int m pal;
                                                  };
};
```

Exercici 2.9.4: Solució posarCartaAPal()

```
bool Joc::posarCartaAPal(int pal)
{ bool movimentFet = false;
   if ( !m destapades.empty())
      if (m_destapades.top().getPal() == pal)
                                                   Cas en que no hi ha cartes:
            (m_pilaResultat[pal].empty())
                                                                   Has de posar l'As
               ( (m_destapades.top().getValor()) == 1)
               movimentFet = true;
               m_pilaResultat[pal].push(m_destapades.top());
               m destapades.pop();
         else
                                                          Cas en que ja hi ha cartes:
                 (m destapades.top().getValor()) ==
                                                                    Has de posar
                  m pilaResultat[pal].top().getValor()+1)
                                                                    carta següent
               movimentFet = true;
               m_pilaResultat[pal].push(m_destapades.top());
               m destapades.pop();
   return movimentFet;
```