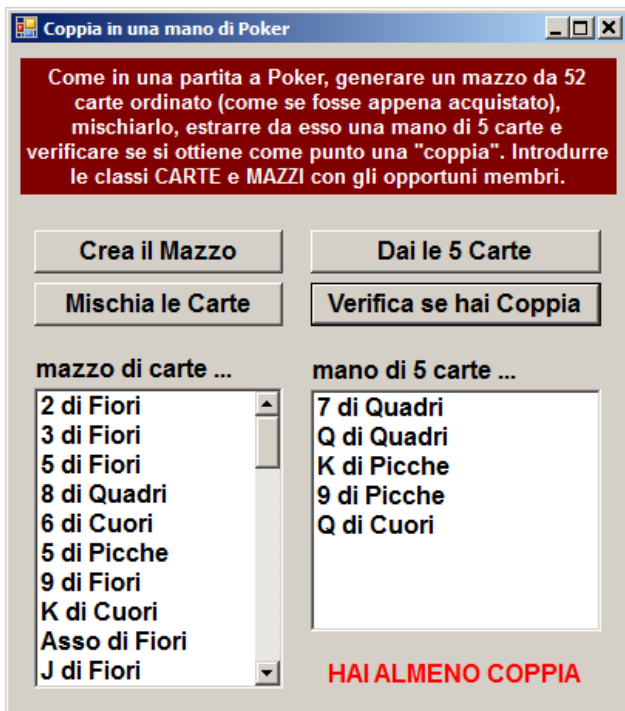


Esempio Pilota:
AlmenoUnaCoppiaInUnaManoDiPoker

Definizione della Form



Codice della Form

```
using System;
using System.Windows.Forms;
```

```
namespace AlmenoUnaCoppiaInUnaManoDiPoker
```

```
{
    public partial class frmAvvio : Form
    {
        public frmAvvio()
        {
            InitializeComponent();
        }
    }
}
```

```
// dichiariamo (ma, per ora, non creiamo) DUE mazzi di carte:
// il mazzo principale e la "mano" del giocatore ...
```

```
Mazzi MazziDiCarte;
Mazzi Mano5Carte;
```

```
private void plsCreaMazzo_Click(object sender, EventArgs e)
{
    MazziDiCarte = new Mazzi(52); // crea un mazzo con 52 carte
    MazziDiCarte.VisualizzaInListBox ( IstMazzo );
}
}
```

```
private void plsMischiaMazzo_Click(object sender, EventArgs e)
{
    if (MazziDiCarte == null)
        MessageBox.Show("Prima di mischiare, CREA il mazzo.");
    else

```

```
{
    MazziDiCarte.Mischia();
    MazziDiCarte.VisualizzaInListBox ( IstMazzo );
}
}
```

```
private void plsDaiLe5Carte_Click(object sender, EventArgs e)
{
    Mano5Carte = new Mazzi (); // ... crea un mazzo vuoto

    for (byte K = 1; K <= 5; K++)
    {
        Carte CartaEstratta = MazziDiCarte.EstraiPrimaCarta();
        Mano5Carte.Aggiungi ( CartaEstratta );
    }

    Mano5Carte.VisualizzaInListBox ( IstMano5Carte );
    MazziDiCarte.VisualizzaInListBox ( IstMazzo );
}
}
```

```
private void plsVerificaCoppia_Click(object sender, EventArgs e)
{
    if ( Mano5Carte.e_Coppia() )
        lblEsito.Text = "HAI ALMENO COPPIA";
    else
        lblEsito.Text = "NON HAI UNA COPPIA";
}
}
```

Codice della Classe Carta

```
using System;
using System.Windows.Forms;
```

```
namespace AlmenoUnaCoppiaInUnaManoDiPoker
```

```
{
    class Carte
    {
        // attributi interni ...

        // i valori che codificano le carte possibili sono:
        // 1 (A), 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 (J), 12 (Q) e 13 (K)

        byte _Numero;

        // i valori che codificano i possibili pali sono:
        // 1 (Cuori), 2 (Quadri), 3 (Fiori), 4 (Picche)

        byte _Palo;

        // costruttori ...
    }
}
```

```
public Carte ()
{
    _Numero = 1; // asso
    _Palo = 1; // di cuori
}
}
```

```
public Carte(byte NumeroDellaCarta, byte PaloDellaCarta)
{
    Numero = NumeroDellaCarta;
    Palo = PaloDellaCarta;
}
}
```

```
// proprietà ...
```

```
public byte Numero
```

```
{
    get { return _Numero; }
    set
    {
        if ((value < 1) || (value > 13))
        {
            MessageBox.Show("Numero Carta errato.");
        }
        else
            _Numero = value;
    }
}
```

```
public byte Palo
```

```
{
    get { return _Palo; }
    set
    {
        if ((value < 1) || (value > 4))
        {
            MessageBox.Show("Numero Palo errato.");
        }
        else
            _Palo = value;
    }
}
```

```
// metodi ...
```

```
public string ToString( )
```

```
{
    return ToStringNumero( ) + " di " + ToStringPalo( );
}
```

```
public string ToStringNumero( )
```

```
{
    if (_Numero == 1)
        return "Asso";
    else if ((_Numero >= 2) && (_Numero <= 10))
        return _Numero.ToString();
    else if (_Numero == 11)
        return "J";
    else if (_Numero == 12)
        return "Q";
    else
        return "K";
}
```

```
public string ToStringPalo( )
```

```
{
    if (_Palo == 1)
        return "Cuori";
    else if (_Palo == 2)
        return "Quadri";
    else if (_Palo == 3)
        return "Fiori";
    else
        return "Picche";
}
```

```
public bool UgualeA (Carte AltraCarta)
```

```
{
    return ((_Numero == AltraCarta.Numero) &&
        (_Palo == AltraCarta.Palo));
}
```

Codice della Classe Mazzi

```
using System;
using System.Collections.Generic; // necessario per usare la classe List
using System.Windows.Forms;
```

```
namespace AlmenoUnaCoppiaInUnaManoDiPoker
```

```
{
    class Mazzi
    {
        // attributi interni ...
        private List<Carte> _Mazzo;
```

```
// costruttori ...
```

```
public Mazzi( )
```

```
{
    _Mazzo = new List<Carte>( );
}
```

```
public Mazzi (byte TipoDiMazzo)
```

```
{
    if (TipoDiMazzo == 52)
    {
        _Mazzo = new List<Carte>( );

        for (byte Kp = 1; Kp <= 4; Kp++)
            for (byte Kn = 1; Kn <= 13; Kn++)
            {
                Carte NuovaCarta = new Carte (Kn, Kp);
                Aggiungi (NuovaCarta);

                // ... si poteva fare direttamente: Aggiungi(new Carte(Kn, Kp))
            }
    }
}
```

```
// metodi ...
```

```
public bool Esiste ( Carte CartaDaVerificare )
```

```
{
    bool Esito = false;

    // scorri tutte le carte del mazzo ...
    for (byte K = 0; K <= _Mazzo.Count - 1; K++)

        // se la carta in posizione K è uguale a quella da verificare ...
        if ( CartaDaVerificare.UgualeA ( _Mazzo [K] ) )
        {
            Esito = true;
            break;
        }

    return Esito;
}
```

```

public void Aggiungi ( Carte NuovaCarta )
{
    if ( Esiste (NuovaCarta) )
    {
        MessageBox.Show(
            "Impossibile aggiungere: carta già presente nel mazzo.");
    }
    else
        _Mazzo.Add ( NuovaCarta );
}

```

```

public Carte EstraiCartaInPosizione ( byte Posizione )
{
    Carte CartaEstratta = _Mazzo [Posizione];
    _Mazzo.RemoveAt (Posizione);

    return CartaEstratta;
}

```

```

public Carte EstraiPrimaCarta()
{
    return EstraiCartaInPosizione ( 0 );
}

```

```

public void Mischia()
{
    Mazzi MazzoMischiato = new Mazzi( );
    Random R = new Random( );
    byte PosizioneCarta = 0;

    // crea un nuovo mazzo estraendo casualmente
    // le carte dal mazzo dell'oggetto ...

    for (int K = 51; K >= 0; K--) // K deve essere int ...alla fine vale -1!
    {
        PosizioneCarta = (byte) R.Next (K + 1);

        MazzoMischiato.Aggiungi
            ( EstraiCartaInPosizione (PosizioneCarta) );
    }

    // azzerare il mazzo dell'oggetto e copia le carte
    // "mischiare" dal nuovo mazzo al mazzo dell'oggetto ...

    _Mazzo = new List<Carte>( );

    for (byte K = 0; K <= 51; K++)
        Aggiungi ( MazzoMischiato.EstraiPrimaCarta() );
}

```

```

public void VisualizzaInListBox(ListBox L)
{
    L.Items.Clear();
    for (byte K = 0; K <= _Mazzo.Count - 1; K++)
        L.Items.Add( _Mazzo[K].ToString() );
}

```

```

public bool e_Coppia()
{
}

```

```

bool Esito = false;

```

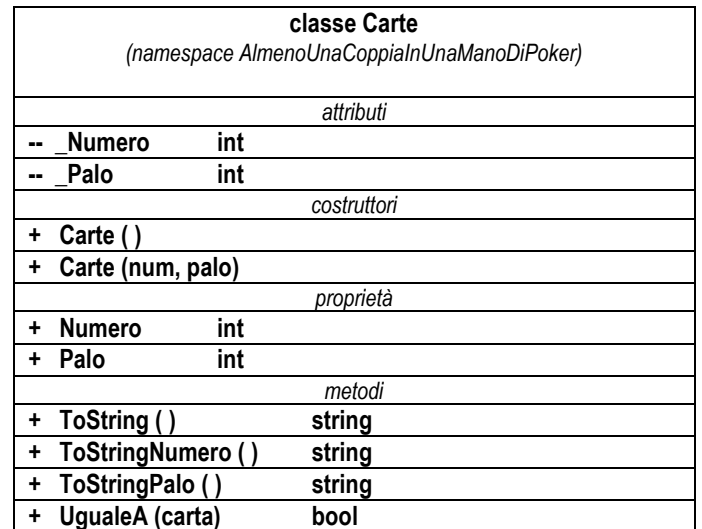
```

for (int K1 = 0; K1 <= _Mazzo.Count - 2; K1++)
    for (int K2 = K1 + 1; K2 <= _Mazzo.Count - 1; K2++)
        if ( _Mazzo [K1].Numero == _Mazzo [K2].Numero)
            Esito = true;

return Esito;
}
}

```

Diagrammi UML delle classi CARTE e MAZZI



Associazione fra classi: MAZZI contengono CARTE

Navigabilità: ogni MAZZO specifica le sue CARTE

