

```

/*
Programma : cppClassPolimorfismo

Autore : Riontino Raffaele 4 inf corso serale
        ITIS Molinari Milano
        7/2/2011

Funizionalità : esempio di polimorfismo delle classi
*/

#include <iostream>
#include <conio.h>
using namespace std;

class Padre
{
public : virtual void Stampa() {cout << "\nPadre\n";};
/*
    il termine virtual permette alla classe (Padre) di poter adottare i metodi
    con lo stesso nome della classe derivata (Figlio). in questo modo è possibile
    poter aggiornare una classe senza doverla riprogettare (riciclo del codice)
*/
};

class Figlio:public Padre
{
public : void Stampa() {cout << "\nFiglio\n";};
};

int main()
{
    Padre *p; //puntatore di tipo padre
    p = new Padre; //associazione del puntatore ad un area di memoria di tipo padre
    p->Stampa(); //utilizzo del metodo stampa (in questo caso verrà utilizzato quello della
    classe padre)
    p = new Figlio; /* se il puntatore viene associato ad un'area di memoria
                     di tipo figlio, i metodi e le proprietà della classe padre che sono precedute
                     dal termine 'virtual' vengono sostituite dai metodi e dalle proprietà,
                     con lo stesso nome, che sono presenti nella classe figlio.
                     */
    p->Stampa(); /*
                 in questo esempio dopo aver associato il puntatore di tipo padre ad un area
                 di memoria di tipo figlio, il metodo stampa utilizzato è quello presente
                 nella classe figlio.
                 */

    getch();
    return 0;
} //fine main

```