

dynamic cast

Il linguaggio di programmazione [C++](#), oltre al [cast](#) esplicito stile C (), dispone di altri quattro tipi di cast espliciti: [const_cast](#), [dynamic_cast](#), [reinterpret_cast](#) e [static_cast](#).

Il costrutto **dynamic_cast** permette di convertire un puntatore ad un oggetto di una classe base ad uno di classe derivata, ma solo se la classe base è [polimorfa](#) (ovvero possiede almeno un metodo virtuale, distruttore virtuale incluso). La validità dell'operazione viene verificata in run-time.

Qualora l'operazione non sia valida il costrutto restituisce un puntatore nullo.

Esempio:

```
class Padre {.....};
class Figlio : public Padre {.....};
class Nipote : public Figlio {.....};

main ()
{
    Padre *p, *pObj = new Padre;
    Figlio *f, *fObj = new Figlio;
    Nipote *n, *nObj = new Nipote;

    p = nObj; // ok un puntatore ad una classe derivata può
essere assegnato ad uno di classe base
    n = p; // qui il compilatore genera errore perché non sa
se l'assegnamento è valido (lo si sa solo run-time)
    n = dynamic_cast<Nipote*>(p); // qui si chiede di fare il controllo run-time
per validare la conversione (valida in questo caso)
    if (n)
        cout << "conversione eseguita con successo";

    p = fObj;
    n = dynamic_cast<Nipote*>(p); // qui si chiede di fare il controllo run-time
per validare la conversione (non valida in questo caso)
    if (!n)
        cout << "conversione fallita";
}
```

Si tenga presente che affinché il costrutto *dynamic_cast* lavori correttamente è necessario che il supporto [RTTI](#) del compilatore sia attivo (spesso, nella produzione del [firmware](#), si sceglie di non attivare [RTTI](#) poiché questa funzionalità assorbe una quantità di risorse non trascurabile).