

# Analisi del codice

Analizziamo il comportamento del codice per determinare le relazioni di sottotipo:

## Prima parte

```
if (transform<A,B>(new C()) == nullptr)
    cout << "Data";
```

Il transform ritorna `nullptr`, quindi stampa "Data". Questo significa che:

- $C \leq A$  (VERO): perché possiamo fare `new C()` e passarlo come `A*`
- $C \leq B$  (FALSO): perché il `dynamic_cast<B*>` fallisce

## Seconda parte

```
if (transform<B,C>(new D()) == nullptr)
    cout << "Structures";
```

Il transform ritorna `nullptr`, quindi stampa "Structures". Questo significa che:

- $D \leq B$  (VERO): perché possiamo fare `new D()` e passarlo come `B*`
- $D \leq C$  (FALSO): perché il `dynamic_cast<C*>` fallisce

## Terza parte

```
if (dynamic_cast<D*>(transform<A,B>(new E()))) != nullptr)
    cout << " and ";
```

La condizione è vera (stampa " and "), quindi:

- $E \leq A$  (VERO): perché possiamo fare `new E()` e passarlo come `A*`
- $E \leq B$  (VERO): perché `transform<A,B>(new E())` non restituisce `nullptr`
- $E \leq D$  (VERO): perché `dynamic_cast<D*>(transform<A,B>(new E()))` non restituisce `nullptr`

## Quarta parte

```
B* pb = transform<A,B>(new F());
if (pb && dynamic_cast<E*>(pb) == nullptr)
    cout << "Algorithms";
```

La condizione è vera (stampa "Algorithms"), quindi:

- $F \leq A$  (VERO): perché possiamo fare `new F()` e passarlo come `A*`
- $F \leq B$  (VERO): perché `transform<A,B>(new F())` non restituisce `nullptr`
- $F \leq \cancel{E}$  (FALSO): perché `dynamic_cast<E*>(pb)` restituisce `nullptr`

## Relazioni di sottotipo

Relazione	Valore
$A \leq B$	FALSO
$A \leq C$	FALSO
$A \leq D$	FALSO
$A \leq E$	FALSO
$A \leq F$	FALSO
$B \leq A$	VERO
$B \leq C$	FALSO
$B \leq D$	FALSO
$B \leq E$	FALSO
$B \leq F$	FALSO
$C \leq A$	VERO
$C \leq B$	FALSO
$C \leq D$	FALSO
$C \leq E$	FALSO
$C \leq F$	FALSO
$D \leq A$	VERO
$D \leq B$	VERO
$D \leq C$	FALSO
$D \leq E$	FALSO
$D \leq F$	FALSO
$E \leq A$	VERO
$E \leq B$	VERO
$E \leq C$	POSSIBILE
$E \leq D$	VERO
$E \leq F$	FALSO
$F \leq A$	VERO
$F \leq B$	VERO

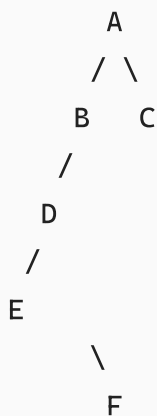
Relazione	Valore
$F \leq C$	POSSIBILE
$F \leq D$	POSSIBILE
$F \leq E$	FALSO

## Giustificazioni principali:

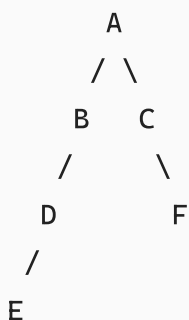
1. A è la radice della gerarchia (tutte le altre classi derivano direttamente o indirettamente da A )
2. B deriva direttamente da A
3. C deriva direttamente da A ma non da B
4. D deriva direttamente da B quindi indirettamente da A
5. E deriva da D quindi indirettamente da B e A
6. F deriva da B (direttamente o indirettamente) ma non da E
7. F potrebbe derivare da C o da D (non possiamo determinarlo con certezza)
8. E potrebbe derivare da C (non possiamo determinarlo con certezza)

## Diagramma della gerarchia

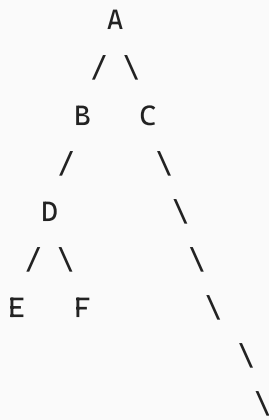
Ecco una possibile gerarchia che soddisfa tutte le condizioni:



Alternativamente, potrebbe essere:



O ancora:



Il punto chiave è che E deriva da D che deriva da B che deriva da A , mentre C deriva direttamente da A , e F deriva da B ma non da E .