

TS = Tipo Statico / TD = Tipo Dinamico

p1->g(); - TS=const B\* / TD=D

- B1 che è const B\* e fa direttamente match con il metodo g() presente nella classe B, essendo che è marcato const e quindi stampa direttamente quello

(p1->n())->g(); - TS=const B\* / TD=D

- Non compila, essendo che il metodo virtual B\* n() prevede un return this, ma il this è costante nel nostro caso; quindi, il metodo compilerebbe se:
  - o Ci fosse virtual B\* n() const
  - o Togliessimo il const con un const\_cast

p2->f(); - TS= B / TD=C

- C'è il metodo f() in B e chiama quello, poi G viene ridefinito dalla sottoclasse correttamente (overriding giusto ed è const) poi stampa B::h essendo presente solo in B
- La stampa successiva (4) fa la stessa cosa, ma con m() (non essendo virtuale) e poi chiama sempre i due stessi metodi

(static\_cast(p2))->k();

- Errore runtime/Undefined behavior = metodo che si chiama ricorsivamente; si può vedere che quando fa questa conversione, tipo dell'espressione va solo dentro D e chiama k() ricorsivamente all'infinito

p3->f();- TS= B / TD=D

- Chiama f() non essendo presente in D, poi corretto overriding (stessa firma/stessi parametri) del metodo dopo chiama h in D (pur non essendoci virtual, basta sia nella superclasse, fa overriding comunque), successivamente B::m non essendo virtuale, rimane in B per g() essendo const e non fa overriding nella sottoclasse, infine stampa h, virtuale ancora una volta perché overriding corretto

Lo stesso dicasi per tutte le stampe successive (in caso di dubbi, chiedete e aggiungo)