



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Polimorfismo: vista d'insieme

Stefano Ghidoni



Agenda

- Riepilogo su polimorfismo
- Analisi del polimorfismo
 - Dinamico
 - Statico
- Confronto tra i polimorfismi
- `dynamic_cast`



Polimorfismo

- Polimorfismo: capacità di associare comportamenti diversi alla stessa notazione generica (Vandevoorde/Josuttis)
- Quali strumenti visti in questo corso causano un comportamento di questo tipo?
 - Ereditarietà + funzioni virtuali
 - Template

Polimorfismo dinamico



Polimorfismo dinamico

- Polimorfismo dinamico: gestito tramite
 - Ereditarietà
 - Funzioni virtuali
- Design:
 - Identificare un insieme comune di funzionalità tra classi collegate
 - Creare il corrispondente insieme di funzioni virtuali nella classe base

Una classe derivata ha un collegamento molto esplicito, a cui deve rispondere, a una classe base

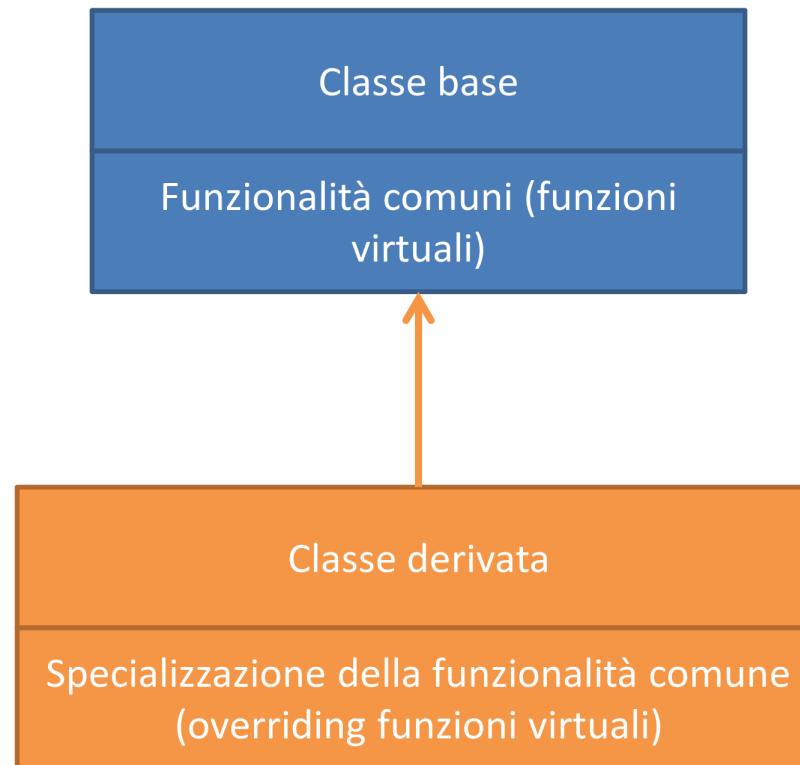


Polimorfismo dinamico

- Esempio classico: varie forme geometriche
 - Stesse operazioni (interfaccia comune nella classe base)
 - Ciascuna operazione eseguita in modo diverso tra una forma e l'altra (ovverriding nella classe derivata)

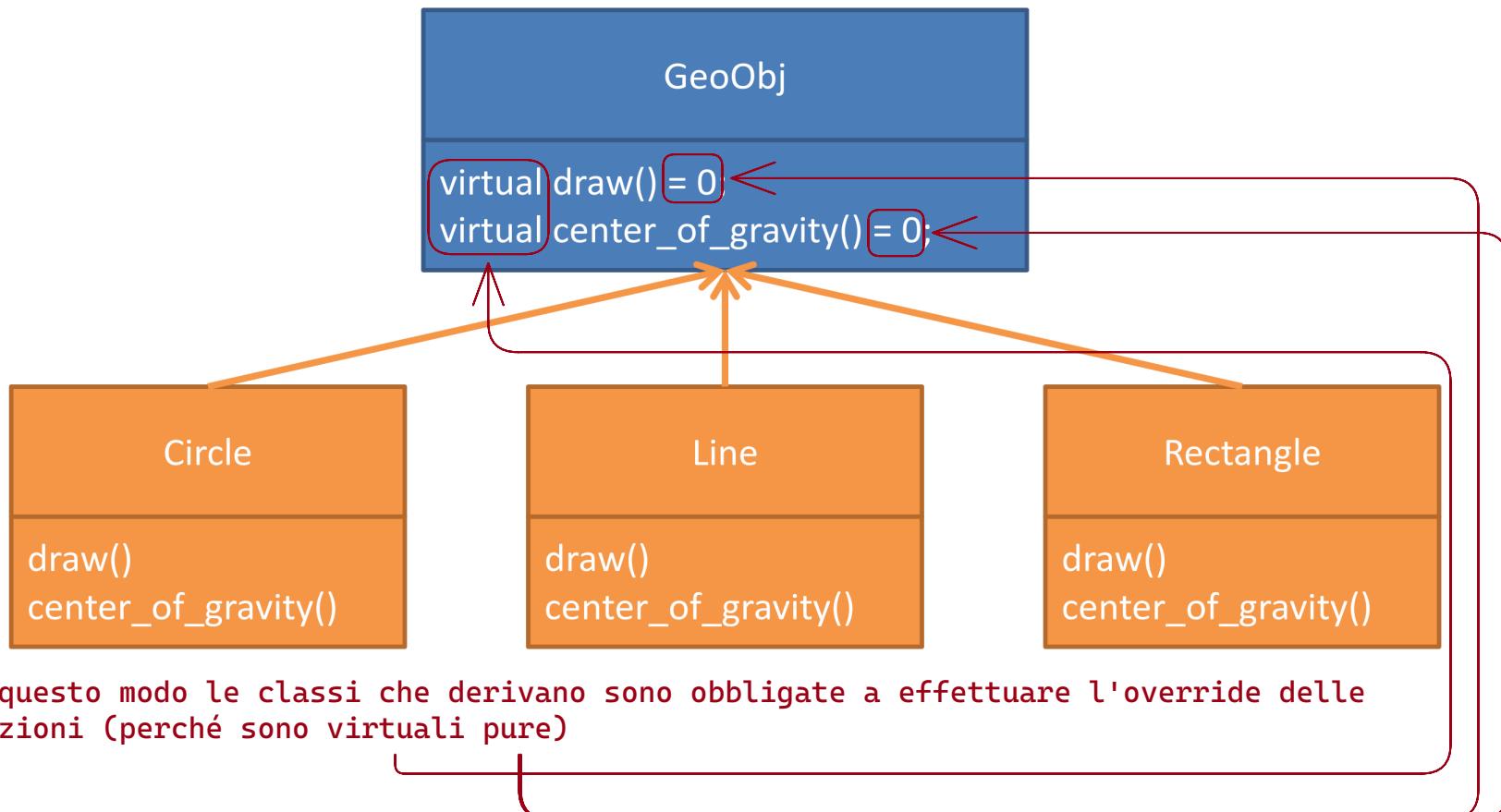


Polimorfismo dinamico





Polimorfismo dinamico



In questo modo le classi che derivano sono obbligate a effettuare l'override delle funzioni (perché sono virtuali pure)



Polimorfismo dinamico

- Chiamata a funzione virtuale

```
void MyDraw(GeoObj const& obj)
{
    obj.draw();           // chiamata alla funzione draw() della
                          // relativa classe
}
```

- È sufficiente una sola funzione **MyDraw()** per disegnare tutte le forme
 - La differenziazione è gestita dalle varie funzioni **draw()**

La separazione avviene sfruttando i virtual pointer e le virtual table, quindi avviene all'interno della gerarchia di classi.

Questo esempio è rappresentazione del polimorfismo: c'è una sola chiamata a una sola funzione, che si può risolvere in più modi diversi.

Polimorfismo statico



Polimorfismo statico

- Presuppone che la stessa sintassi sia condivisa dalle varie forme (cioè dalle varie istanze del template)
- Le classi / funzioni sono definite in maniera indipendente
 - Ma devono poter gestire la stessa sintassi
- Il polimorfismo si concretizza specificando la classe / funzione con vari tipi diversi



Polimorfismo statico

- Circle, Line e Rectangle hanno le stesse funzioni

→ Non le implementano per obbligo (come nell'esempio del polimorfismo dinamico).

Circle

```
draw()  
center_of_gravity()
```

Line

```
draw()  
center_of_gravity()
```

Rectangle

```
draw()  
center_of_gravity()
```



Polimorfismo statico

- Chiamata a funzione template

```
template <typename GeoObj>
void MyDraw(GeoObj const& obj)
{
    obj.draw();           // chiamata alla funzione draw() della
                          // classe specificata nel template
}
```

- Quante funzioni MyDraw() esistono?

- Ne esistono molte:
 - MyDraw<Circle>()
 - MyDraw<Line>()
 - MyDraw<Rectangle>()

Prima un'interfaccia comune obbligava le classi derivate a implementarla, ora l'obbligo è implicito (serve una funzione draw() per poter compilare la funzione MyDraw()).
Formalmente, la funzione MyDraw è unica, poi il compilatore specializza per il tipo specifico.



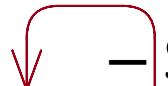
Polimorfismo dinamico vs statico

- Polimorfismo tramite ereditarietà:
 - Vincolato (bounded): l'interfaccia delle classi derivate è vincolata dalla classe base
 - Dinamico: binding effettuato a run-time (dinamicamente)



Polimorfismo dinamico vs statico

- Polimorfismo tramite template:



In realtà, ora
non derivano da
nulla, quindi
l'uso del termine
è improprio

- Svincolato (unbounded): l'interfaccia delle classi derivate non è predeterminata (non esiste il corrispondente di una classe base)
- Statico: binding effettuato a tempo di compilazione (staticamente)



Polimorfismo – definizione

- Definizione di polimorfismo:
 - **Polimorfismo dinamico**
 - Implementato tramite ereditarietà e funzioni virtuali
 - A tempo di esecuzione (run-time)
 - **Polimorfismo statico**
 - Implementato tramite template
 - A tempo di compilazione (compile-time)



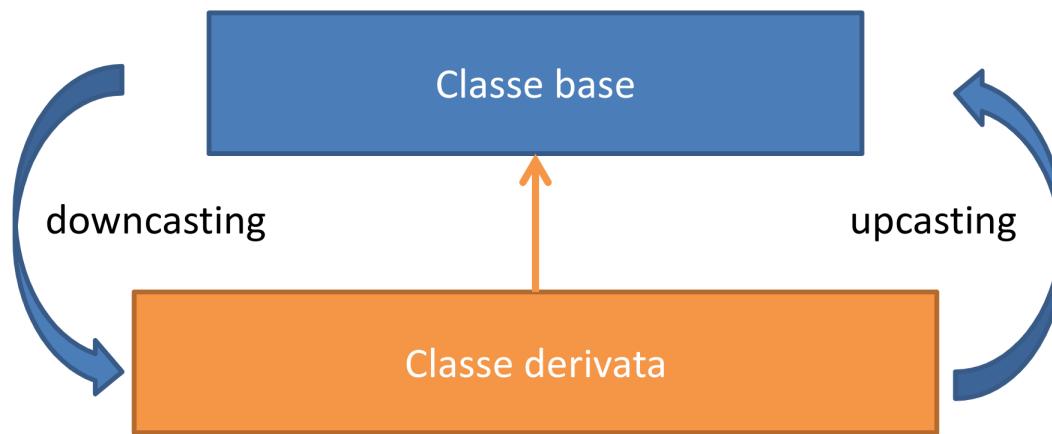
Altre forme di polimorfismo

- Anche altri meccanismi possono essere visti come polimorfismo:
 - Overloading di funzioni
 - Overloading di operatori



dynamic_cast

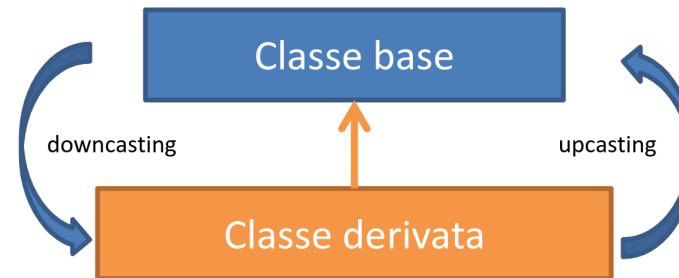
- `dynamic_cast` è utilizzato per "navigare" nella gerarchia di classi





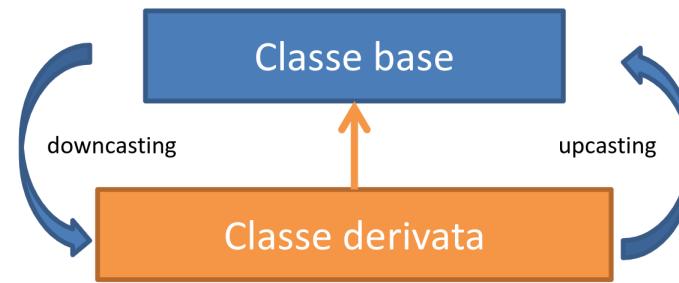
Upcasting

- Upcasting è sempre lecito
 - Un oggetto di classe derivata è un (*is a*) oggetto di classe base
 - Non necessita di cast



Downcasting

- Downcasting: effettuato con `dynamic_cast`
- Funziona con i puntatori e con le reference
- **Unico cast che effettua verifiche a run-time**
- Se il cast non è lecito:
 - Puntatori: ritorna `nullptr`
 - Reference: lancia un'eccezione (`bad_cast`)



Bisogna sempre controllare
che il downcast sia lecito
dopo averlo effettuato



Recap

- Analisi e definizione di polimorfismo
- Dettaglio e confronto dei polimorfismi
 - Dinamico
 - Statico
- Dove si genera il polimorfismo
- Altre forme di polimorfismo
- Upcasting, downcasting e dynamic_cast



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Polimorfismo: vista d'insieme

Stefano Ghidoni