10.3 – Ereditarietà: Memory layout degli oggetti

Libro di testo:

- Capitoli 14.3.1

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Agenda

- Layout della memoria degli oggetti con classi derivate
- Virtual pointer
- Virtual table

Layout degli oggetti

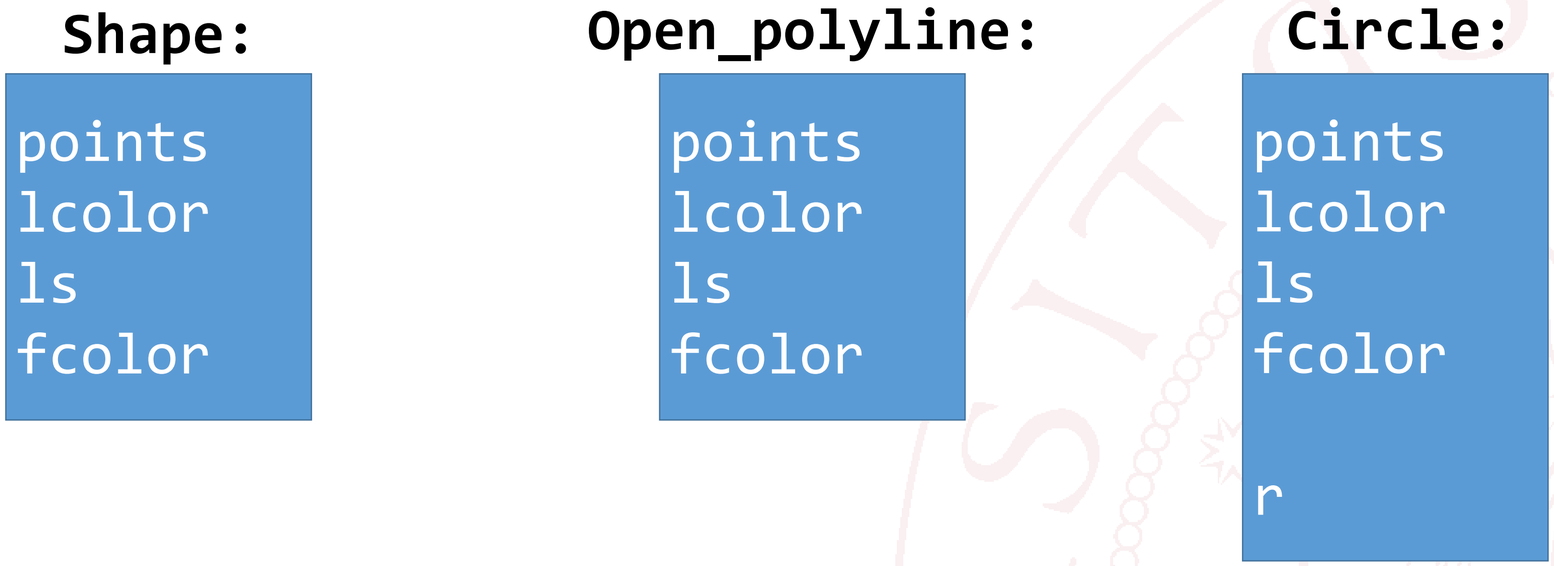
- Recall: la classe base `Shape` contiene

```
    std::vector<Point> points;   // non usato da tutte le Shape
    Color lcolor;                // dalla libreria grafica
    Line_style ls;               // dalla libreria grafica
    Color fcolor;                // dalla libreria grafica
```

- Consideriamo due classi derivate di `Shape`:
  - `Open_polyline`
    - Nessun dato membro aggiuntivo, si appoggia su `points`
  - `Circle`
    - Dato membro aggiuntivo: il raggio (`r`)

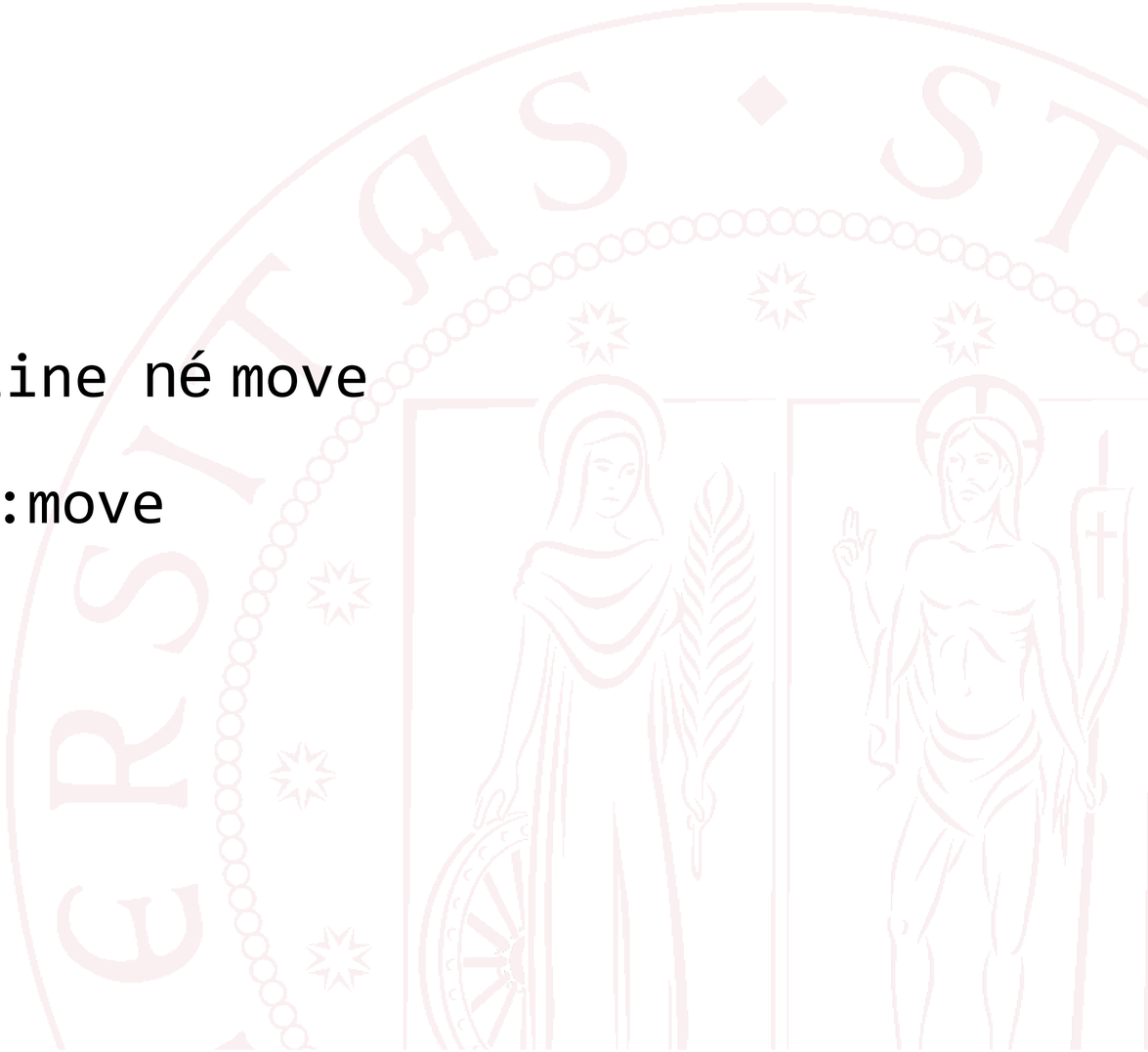
Memory layout degli oggetti derivati (dati membro)

- I dati membro di una classe derivata sono aggiunti dopo quelli della classe base



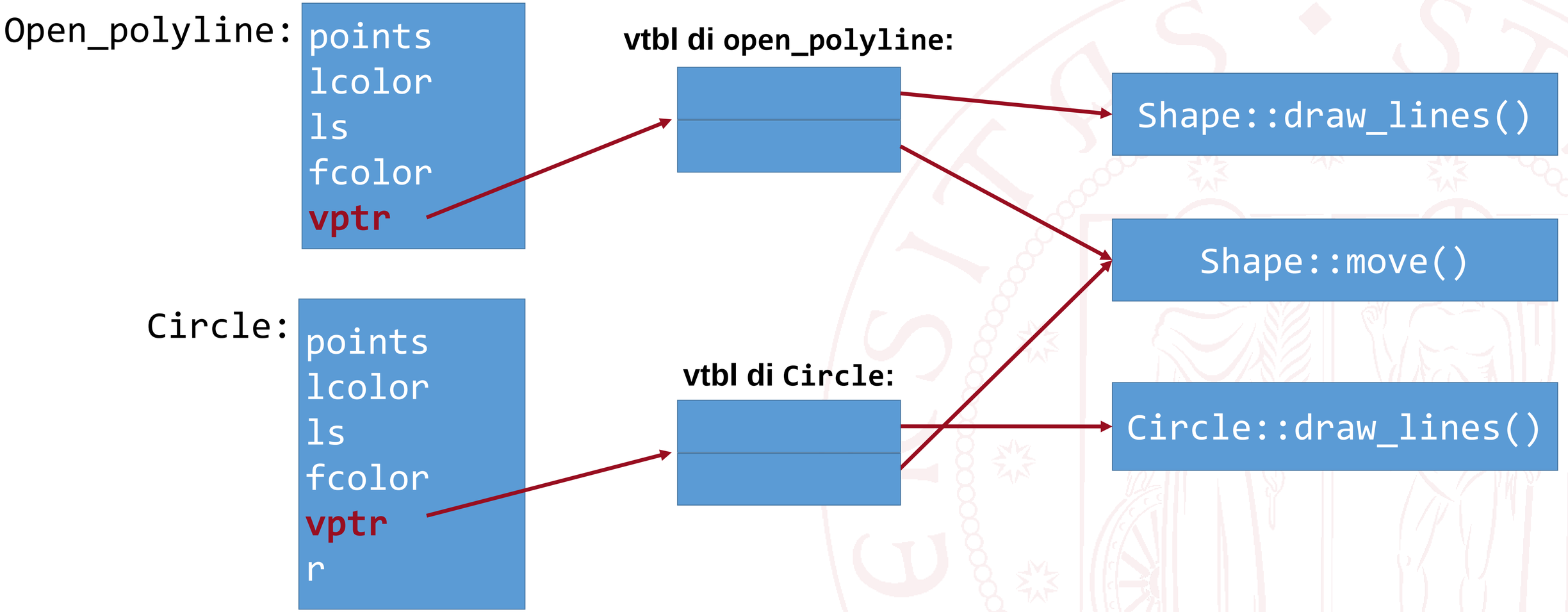
Memory layout degli oggetti derivati (funzioni membro)

- Una chiamata a **funzione virtuale** è gestita tramite le **virtual table (vtbl)**
- **vtbl** gestite tramite **vptr** in ogni oggetto
- Nel nostro caso:
  - `Open_polyline` non ridefinisce né `draw_line` né `move`
  - Ma chiama `Shape::draw_line` o `Shape::move`
  - `Circle` ridefinisce solo `draw_lines`



vtbl e vptr

- Per ogni classe (non ogni oggetto!) è definita una tabella che definisce quali funzioni devono essere chiamate



Recap

- Layout degli oggetti con classi derivate
- Virtual pointer
- Virtual table