

PROGRAMMAZIONE → SOLI O PRINCIPI → best practice per scrivere
del codice mantenibile
↳ 5 PRINCIPI

4 consigli di Carolyn di libri da leggere

- REFACTORING Fowler → ha scritto i principi Agile

- CLEAN CODE Martin

↳ Il codice si deve spiegare da solo, non è una buona idea mettere un commento il fare per spiegare il fare non FUNZIONA piuttosto riscrivere il codice

- TOO Kent Beck

- EFFECTIVE JAVA J. Block

SINGLE RESPONSIBILITY PRINCIPLE → UNA CLASSE O OBIETTO
AVERE SOLO UNA RAGIONE
PER CAMBIARE

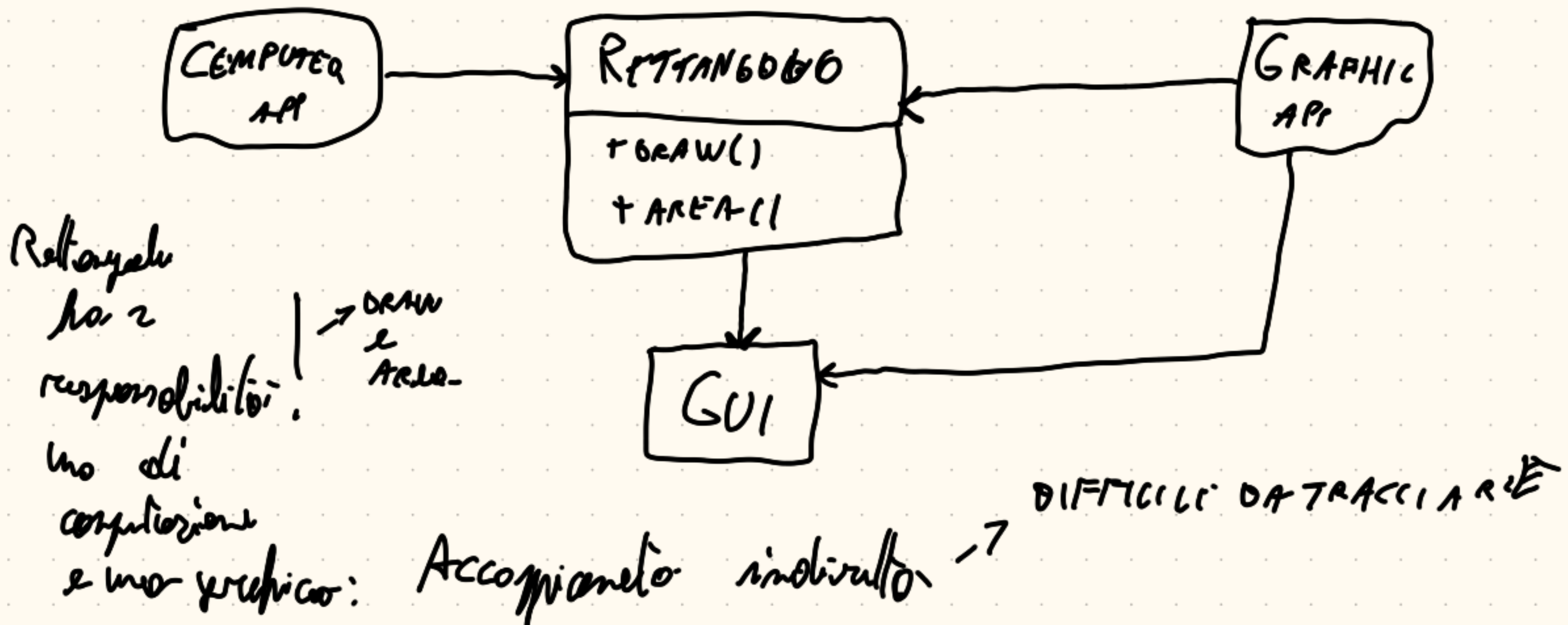
- Also known as cohesion

non possono essere suddiviso in modulo

- La cohesion è alla base della Programmazione orientata agli oggetti

OGGETTO: un insieme di funzioni che insistono tutti sugli stessi dati!

Scala: linguaggio super espressivo ma crea grandi guai



OPEN-CLOSE Principle (non può essere soddisfatto in modo assoluto)

// esempio del nuovo stop

Non devo modificare il code ma solo aggiungere in nuovo-
tipo-

NON è OPEN-CLOSE se faccio-

→ OBLIGATORI

- CREARE

- QUANTO

- TRIMINGOLO

NON è OPEN-CLOSE

→ BISOGNA

↳ SO SO CLASSE ✓

→ Obiettivo FINE: far sì che i metodi siano compatibili con solo-
la firma del metodo

⚠ Se un metodo ha un nome troppo lungo e fa 2 cose
allora è meglio separare i metodi.

LISPOV SUBSTITUTION PRINCIPLE