

## Esercitazione 3



Planning & Scheduling Team  
ISTC-CNR

## Il mondo delle scatole e dei container

Nel *mondo dei container* esistono solo due tipi di entità: **scatole** e **container**. Una scatola ha tre dimensioni intere, ed ogni scatola ha una etichetta identificativa. I container possono contenere scatole. Inoltre, ogni container ha una sua capienza finita, e dunque può contenere solo un numero limitato di scatole.

- **Esercizio 1.** Definire la classe Scatola tale che si possano creare scatole a partire dalle loro dimensioni e dall'etichetta. Definire, inoltre, metodi per ottenere le dimensioni di una scatola ed un metodo per restituire il volume di una scatola.
- **Esercizio 2.** Definire il metodo toString() della classe Scatola per la stampa degli attributi di una scatola (etichetta, dimensioni, volume).
- **Esercizio 3.** Definire la classe Container tale che si possano creare container specificandone il volume. Definire il metodo per inserire una scatola in un container. Aggiungere una scatola deve essere possibile solo se non si eccede la capienza del container.
- **Esercizio 4.** Definire il metodo toString() della classe Container per la stampa degli attributi di un container (capienza, scatole contenute).



Planning & Scheduling Team  
ISTC-CNR

## Il mondo delle scatole e dei container

- **Esercizio 5.** Notiamo che nel mondo dei container esistono delle scatole particolari: le scatole *cubiche* (con dimensioni tutte uguali). Definire la classe `ScatolaCubica` come sottoclasse della classe `Scatola`.
- **Esercizio 6.** Aggiungere alla classe `Container` un metodo per stabilire se esso contiene una scatola identificata dall'etichetta.
- **Esercizio 7.** Aggiungere alla classe `Container` un metodo per rimuovere una scatola identificata dall'etichetta. Se non esiste nel container una scatola identificata dall'etichetta fornita, il metodo non fa nulla.
- **Esercizio 8.** Sono stati inventati dei container "intelligenti" che non permettono di inserire al loro interno scatole omonime. Definire la classe `ContainerIntelligente` come sottoclasse di `Container`.

