

# Metodologia Top-down

Contenuti rieditati delle slide della  
Prof. L. Caponetti

# Metodologia Top-down

- La metodologia **top-down** di sviluppo di un programma riguarda i metodi di risoluzione di un problema per analisi, basati sulla **decomposizione del problema**.
- **Stepwise refinement – raffinamento per passi successivi** – è una strategia top-down proposta da Wirth nell'ambito della programmazione strutturata, nell'articolo «*Program Development by Stepwise Refinement*», 1971

# Metodologia top-down

- La metodologia **top-down** è utilizzata per lo sviluppo di programmi di dimensioni limitate

# Metodologia top-down nell'analisi del problema

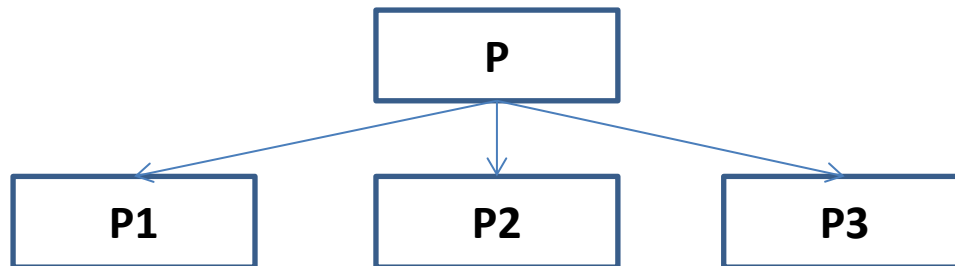
- L'analisi **top-down** prevede la scomposizione del problema generale da risolvere in più **sotto-problemi** a loro volta da risolvere per ottenere la soluzione del problema originario
- Ciascuno di tali **sotto-problemi** viene poi affrontato, come il problema originario, identificando un insieme di sotto-problemi la cui soluzione determina la risoluzione del problema originario

# Metodologia top-down

- Il procedimento di scomposizioni successive viene iterato fino a che tutti i sotto-problemi individuati siano **facilmente risolvibili** nel linguaggio di programmazione scelto
- Il problema iniziale viene progressivamente decomposto in sotto-problemi più semplici che possono essere rappresentati in un **albero** detto di **decomposizione funzionale**

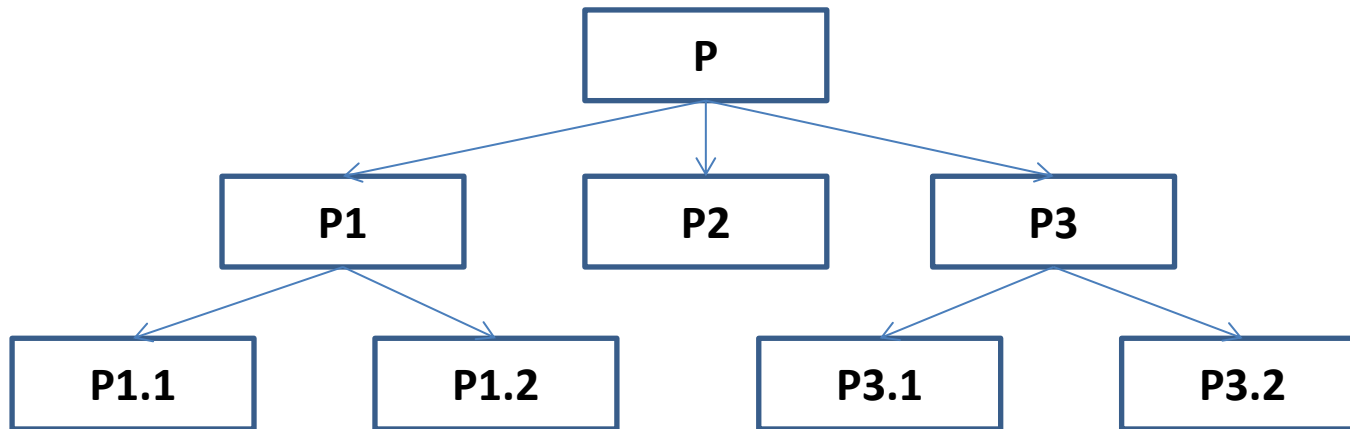
# Albero di decomposizione

- Il problema iniziale è decomposto in un insieme di problemi più semplici
- Ad esempio, il problema iniziale **P** è decomposto nei sotto-problemi **P1**, **P2** e **P3**



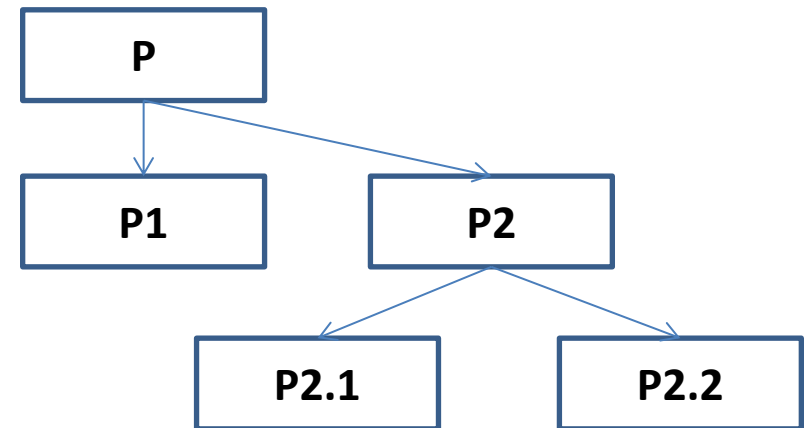
# Albero di decomposizione

- Se i sotto-problemi individuati sono ancora complessi, ciascuno di essi può essere ancora decomposto



# Albero di decomposizione

- La decomposizione può essere rappresentata mediante un albero



- Nella radice **P** c'è il problema iniziale
- Nelle foglie **P1**, **P2.1**, **P2.2** ci sono i sotto-problemi elementari, considerati facilmente risolvibili



# Come comporre le soluzioni dei sotto-problemi

- Lo scopo del processo di decomposizione è di trasformare la soluzione del problema iniziale nella soluzione di molti problemi più semplici e più facilmente risolvibili
- La soluzione per un problema **P** si ottiene componendo le soluzioni dei suoi sotto-problemi utilizzando le strutture di controllo di **sequenza**, **selezione** e **iterazione**
- Un caso a parte si ha quando un problema ammette una soluzione **ricorsiva**

# Specifica di un problema

- La specifica di un problema è data da:
  - Il compito o *funzione* che il problema richiede sia svolto
  - I dati di *input* e *output*

# Soluzione

- La descrizione della soluzione per il problema **P** è basata sulla specifica dei suoi sotto-problemi, cioè:
  - Che cosa si richiede a ciascun sotto-problema **P1**, **P2**, ... (cioè la funzione o compito)
  - Quali sono i dati di ingresso e uscita di ciascun sotto-problema
- In modo analogo per la descrizione di ciascun sotto-problema

# Linee guida

- Nel costruire la soluzione del problema generale si suppone che i sotto-problemi siano stati già risolti
- Con la stessa tecnica si affronta ciascun sotto-problema

# Metodologia bottom-up

- La metodologia **bottom-up** di progettazione e sviluppo di un programma riguarda i metodi di risoluzione di un problema per sintesi, basata sulla **interconnessione di componenti già disponibili**
- Questa metodologia porta alla costruzione di componenti standard riutilizzabili

# Programmazione in grande e in piccolo

- Compito della **programmazione in grande** è il progetto della struttura del programma come un insieme di parti (**moduli**)
- La **programmazione in piccolo** si occupa del progetto e della realizzazione di ciascuna parte, mediante il progetto e l'implementazione di algoritmi

# Progetto di un algoritmo

- Come si arriva alla stesura di un algoritmo in un linguaggio di programmazione?
- Si utilizza una metodologia **top-down** o per **raffinamenti successivi – stepwise refinement**

# Stepwise refinement

- Si intende la **tecnica di sviluppare diverse versioni di un algoritmo** aumentando gradualmente il livello di dettaglio e di formalità del codice
- Inizialmente si producono formulazioni astratte che poi vengono via via raffinate precisando **operazioni, controllo e dati**



# Pseudo-codifica

- Il linguaggio utilizzato nelle diverse formulazioni viene trasformato gradualmente da **linguaggio naturale** a **linguaggio di programmazione** attraverso versioni miste (**pseudocodice**)

# Livelli di astrazione

Algoritmo espresso in linguaggio naturale – **TOP**

Pseudo-codice

Pseudo-codice

Codice

- **DOWN**

# Livelli di astrazione

- L'algoritmo è espresso inizialmente in modo astratto in **linguaggio naturale** (mediante l'uso di astrazioni sul controllo, sui tipi di dato e sulle funzioni)
- Tale descrizione è raffinata (dettagliata) in diversi passi, producendo descrizioni in **pseudo-codice**
- Il processo di raffinamento termina quando l'algoritmo è espresso in modo da essere codificato in un **linguaggio di programmazione** (procedurale cioè algoritmico)