

Programmazione Procedurale

Concetti Introduttivi

Linguaggi di Programmazione

versione 4.1

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons  
(vedi ultima pagina)

creative commons

Concetti Introduttivi: Linguaggi >> Sommario

## Sommario

- Linguaggi di Programmazione
  - ⇒ Sintassi e Semantica
- Il Processo di Sviluppo del Codice
- Compilazione del Codice
- Interpretazione del Codice

Concetti Introduttivi: Linguaggi >> Linguaggi di Programmazione

## Linguaggi di Programmazione

- Servono a scrivere programmi
  - ⇒ ogni programma è una sequenza di istruzioni
  - ⇒ normalmente pensato per risolvere un problema di calcolo
  - ⇒ al programma vengono forniti dei dati
  - ⇒ il programma calcola eseguendo le istruzioni
  - ⇒ il programma restituisce i risultati

Concetti Introduttivi: Linguaggi >> Linguaggi di Programmazione

## Evoluzione dei Linguaggi

- Esistono numerosi linguaggi
  - ⇒ differenti per funzionalità e tecnologia

anni '60	metà anni '60	1974	1981
COBOL	BASIC	C	Obj-C
1968		1984	
Pascal		Matlab	
1990			
C++		1991 VB, Python	
2000			
FORTRAN		C#, VB.NET	
Linguaggi di Programmazione Procedurale		Linguaggi Ibridi	
1995 Java JavaScript		2010> Swift Dart	
Linguaggi Orientati agli Oggetti			

Concetti Introduttivi: Linguaggi >> Linguaggi di Programmazione

## Evoluzione dei Linguaggi

- Prog. Procedurale
  - ⇒ un'applicazione è composta da un insieme di sottoprogrammi
  - ⇒ sottoprogramma = blocco di istruzioni
  - ⇒ l'esecuzione corrisponde all'esecuzione dei sottoprogrammi

```

graph TD
    Programma[Programma] --> stampa[stampa]
    Programma --> calcolaArea[calcolaArea]
    Programma --> main[main]
    stampa --> Programma
    calcolaArea --> Programma
    main --> Programma
  
```

Concetti Introduttivi: Linguaggi >> Linguaggi di Programmazione

## Evoluzione dei Linguaggi

- Prog. Orientata agli Oggetti
  - ⇒ un'applicazione è composta da un insieme di componenti
  - ⇒ che si chiamano classi oppure oggetti
  - ⇒ che si scambiano tra di loro messaggi
  - ⇒ un componente chiede ad un altro di eseguire un'operazione

Concetti Introduttivi: Linguaggi >> Linguaggi di Programmazione

## Evoluzione dei Linguaggi

The diagram shows a 'utente' (user) interacting with a system composed of four 'Componente' (component) boxes. Each component box contains 'informazioni' (information) and 'servizi' (services). The components are interconnected by 'messaggio' (message) arrows. Gears are shown above the components, indicating the flow of data and the evolution of the system.

Concetti Introduttivi: Linguaggi >> Linguaggi di Programmazione

## Linguaggi di Programmazione

The diagram compares two paradigms:

- programmazione procedurale:** Represented by a box containing 'C'.
- programmazione orientata agli oggetti:** Represented by a box containing '“clean C”'.

An arrow points from 'C' to '“clean C”' with the text: 'linguaggio di questo corso - più semplice del C' (language of this course - simpler than C).

**ATTENZIONE** (ATTENTION) points to '“clean C”' with the text: 'ai linguaggi di riferimento' (to the reference languages).

Other boxes represent modern languages: 'C++', 'Java, Python C#', and 'JavaScript'.

## Linguaggi di Programmazione

- Attenzione
  - ⇒ i linguaggi della stessa famiglia sono normalmente basati sugli stessi concetti
- Differenze
  - ⇒ alcuni linguaggi forniscono funzionalità che altri non hanno
  - ⇒ “sintassi”
  - ⇒ “semantica”

## Sintassi e Semantica

- Simile al linguaggio naturale
  - ⇒ sintassi e semantica
- Sintassi
  - ⇒ “grammatica” del linguaggio
  - ⇒ insieme delle regole che stabiliscono quali frasi (programmi) sono corretti
  - ⇒ in Italiano: “maiuscola dopo il punto”
  - ⇒ in C++: “le istruzioni si concludono con ;”

Concetti Introduttivi: Linguaggi >> Linguaggi di Programmazione

## Sintassi e Semantica

- Semantica
  - ⇒ “significato” del linguaggio
  - ⇒ insieme delle regole che stabiliscono come il calcolatore esegue i programmi corretti
  - ⇒ in italiano: “andrò è un’azione nel futuro”
  - ⇒ in C++: `cout << "Ciao";`
- Attenzione
  - ⇒ sintassi e semantica dei linguaggi di programmazione sono rigorose

Concetti Introduttivi: Linguaggi >> Linguaggi di Programmazione

## Sintassi e Semantica

- Attenzione
  - ⇒ ci sono due diverse nozioni di correttezza
- Correttezza Sintattica
  - ⇒ assenza di errori sintattici
  - ⇒ il programma è eseguibile
- Correttezza Semantica (o Logica)
  - ⇒ implica la correttezza sintattica
  - ⇒ il prog. risolve correttamente il problema



## Sintassi e Semantica

- Verifica della correttezza sintattica
  - ⇒ viene verificata da uno strumento apposito, compilatore oppure interprete (>>)
  - ⇒ consente di eliminare gli errori sintattici
- Verifica della correttezza semantica
  - ⇒ esecuzione e “test” del programma
  - ⇒ prove di funzionamento per controllare che il programma si comporti correttamente (processo più complesso e delicato)

## Il Processo di Sviluppo

- Il punto di partenza
  - ⇒ la descrizione del problema, normalmente fornita in linguaggio naturale (“specifica”)
  - ⇒ è opportuno analizzarla accuratamente
- Il punto di arrivo
  - ⇒ l’applicazione correttamente funzionante
- Un passo intermedio fondamentale
  - ⇒ concepire una strategia per la soluzione del problema

Concetti Introduttivi: Linguaggi >> Il Processo di Sviluppo

## Il Processo di Sviluppo

- Algoritmo
  - ⇒ strategia per la soluzione del problema
- Il problema centrale
  - ⇒ capito il problema, decidere attraverso quale sequenza di passi costruire la soluzione
  - ⇒ è un problema ricorrente, che capita in molti altri contesti

Concetti Introduttivi: Linguaggi >> Il Processo di Sviluppo

## Il Processo di Sviluppo

- Avendo concepito l'algoritmo
  - ⇒ è possibile procedere alla scrittura del codice
- Codice sorgente
  - ⇒ istruzioni del linguaggio di programmazione
  - ⇒ non è direttamente eseguibile dal processore
- Linguaggio macchina
  - ⇒ linguaggio di comandi eseguibili dal processore (molto semplici)

Concetti Introduttivi: Linguaggi >> Il Processo di Sviluppo

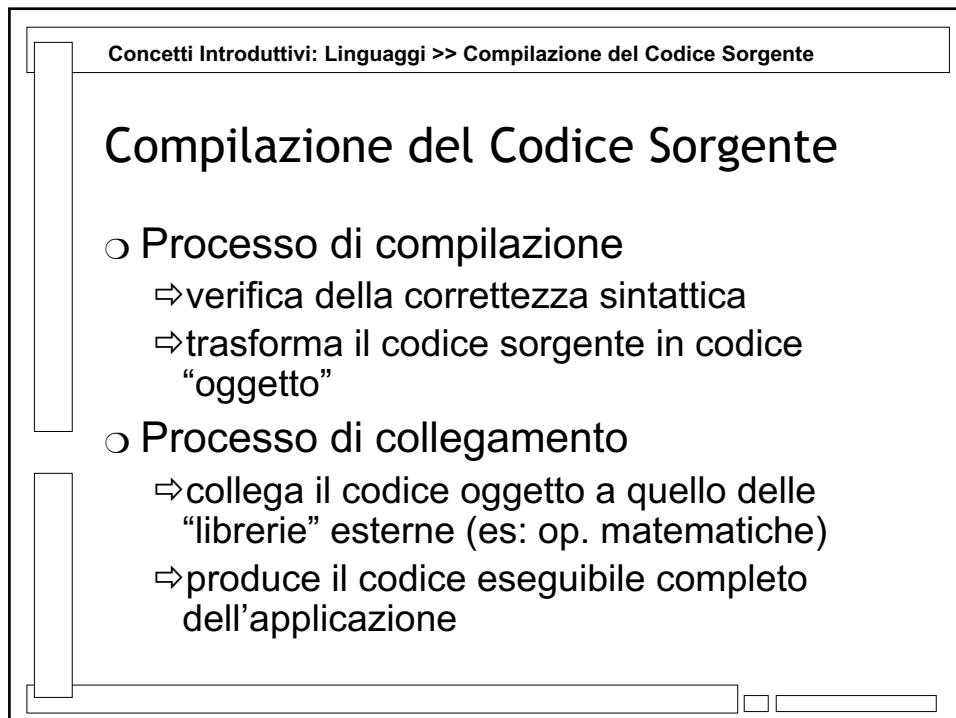
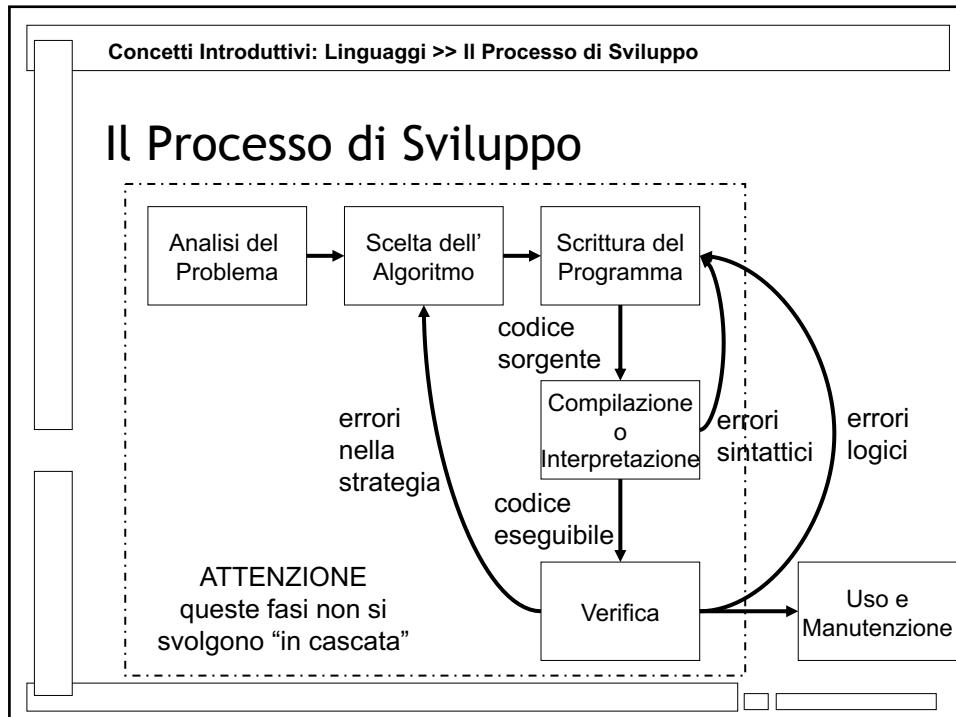
## Il Processo di Sviluppo

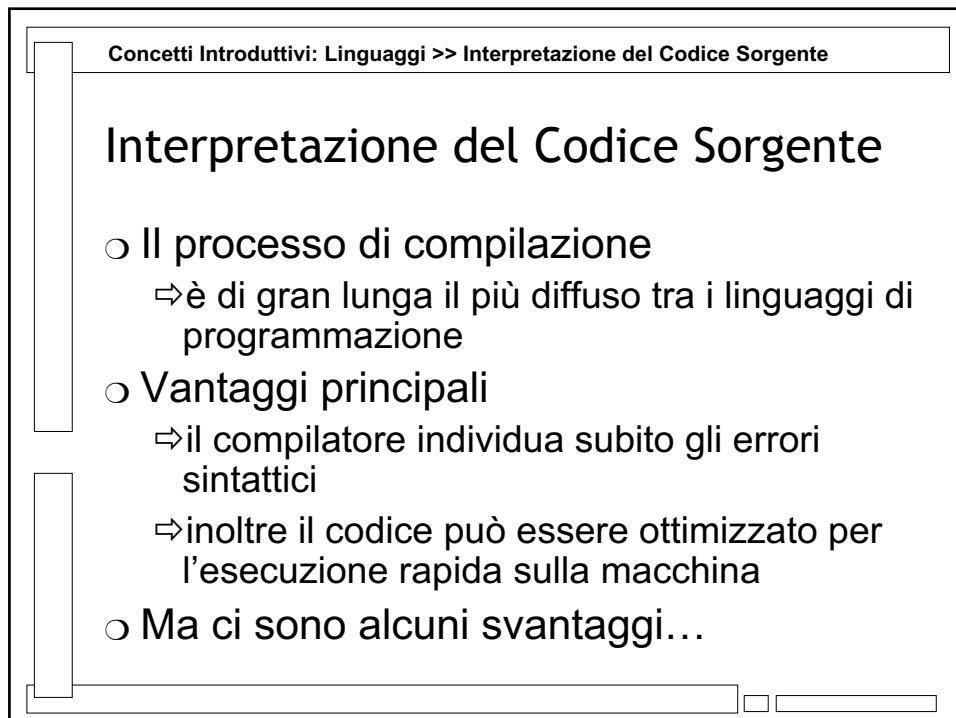
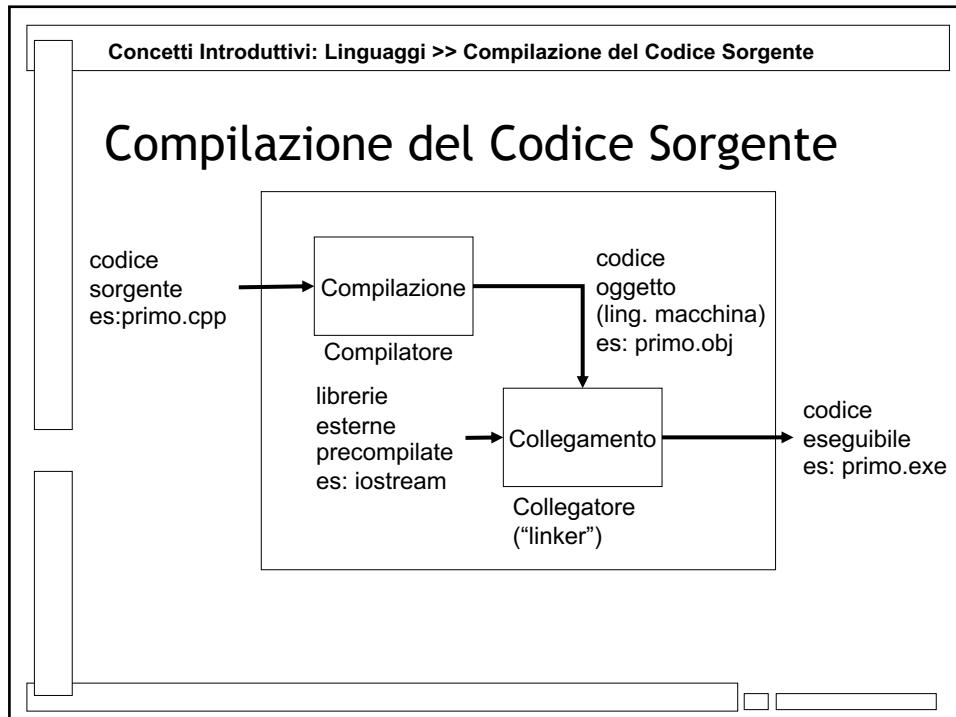
- Un passo cruciale
  - ⇒ la trasformazione del codice sorgente in codice macchina
- Esistono due approcci
  - ⇒ compilazione del codice sorgente – richiede l'utilizzo di uno strumento chiamato compilatore
  - ⇒ interpretazione del codice sorgente – richiede l'utilizzo di uno strumento chiamato interprete

Concetti Introduttivi: Linguaggi >> Il Processo di Sviluppo

## Il Processo di Sviluppo

- A questo punto
  - ⇒ l'applicazione è eseguibile
  - ⇒ è necessario verificarne la correttezza logica
- Fase di test
  - ⇒ esecuzione ripetuta dell'applicazione su dati di test
  - ⇒ per verificarne il funzionamento





## Interpretazione del Codice Sorgente

- Svantaggio principale
  - ⇒ il processo di verifica (test) del codice è meno immediato
- Infatti non è possibile
  - ⇒ scrivere il codice e schiacciare un tasto per eseguirlo
  - ⇒ è sempre prima necessario eseguire passi intermedi (salvare il codice in un file, eseguire il compilatore, e solo dopo la compilazione eseguire il file eseguibile)

## Interpretazione del Codice Sorgente

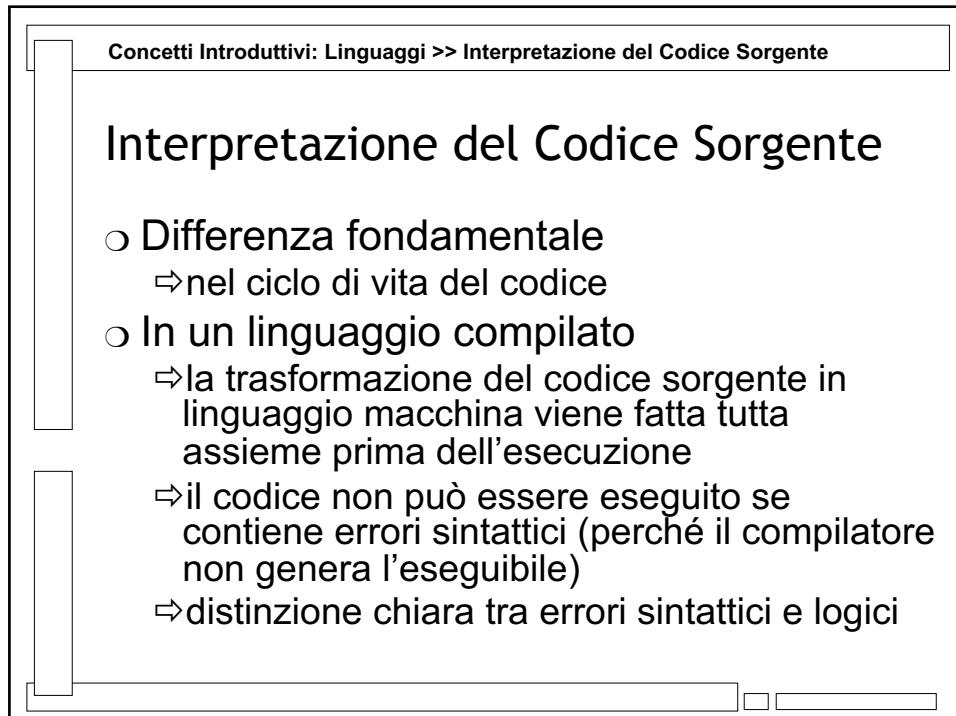
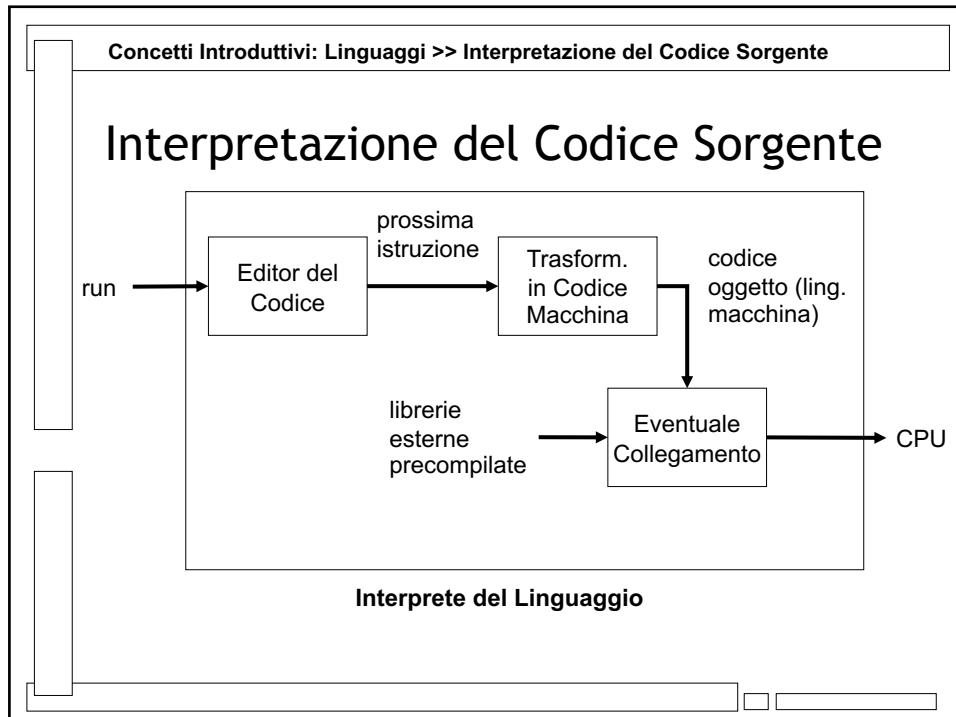
- Un processo molto più immediato
  - ⇒ interpretazione del codice
- Idea
  - ⇒ il codice viene scritto all'interno di uno strumento apposito, detto interprete
  - ⇒ che ovviamente supporta anche le funzioni di scrittura ("editing" di testo)
  - ⇒ inoltre fornisce un comando "Esegui" oppure "Run"

## Interpretazione del Codice Sorgente

- All'esecuzione del comando “Run”
  - ⇒ l'interprete analizza le istruzioni una per una, a partire dalla prima
  - ⇒ traduce ciascuna istruzione in linguaggio macchina e chiede al processore di eseguirla
  - ⇒ poi passa all'istruzione successiva
- In altri termini
  - ⇒ non è necessario salvare il codice in un file
  - ⇒ non è necessario eseguire il comando di compilazione prima dell'esecuzione

## Interpretazione del Codice Sorgente

- Il collegamento
  - ⇒ avviene tipicamente in modo dinamico, ovvero ogni volta che un'istruzione del codice lo richiede
  - ⇒ al momento dell'esecuzione di quella istruzione, l'interprete localizza la libreria esterna
  - ⇒ e la collega al codice dell'applicazione per eseguirla



Concetti Introduttivi: Linguaggi >> Interpretazione del Codice Sorgente

## Interpretazione del Codice Sorgente

- In un linguaggio interpretato
  - ⇒ la trasformazione del codice sorgente in linguaggio macchina viene fatta istruzione per istruzione durante l'esecuzione
  - ⇒ il codice può essere eseguito anche se contiene errori sintattici
  - ⇒ in questo caso l'esecuzione comincia e poi si interrompe al momento in cui viene incontrata l'istruzione sintatticamente scorretta

Concetti Introduttivi: Linguaggi >> Sommario

## Riassumendo

- Varie tecnologie per la programmazione
  - ⇒ ci concentriamo sulla prog. procedurale
  - ⇒ sintassi e semantica
- Creazione di un Programma
- Compilazione e Interpretazione
- Metodologia di Programmazione

Termini della Licenza

## Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza “Attribution-ShareAlike” di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all’indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Concetti Introduttivi: Linguaggi >> Il Concetto di Algoritmo

## Il Concetto di Algoritmo

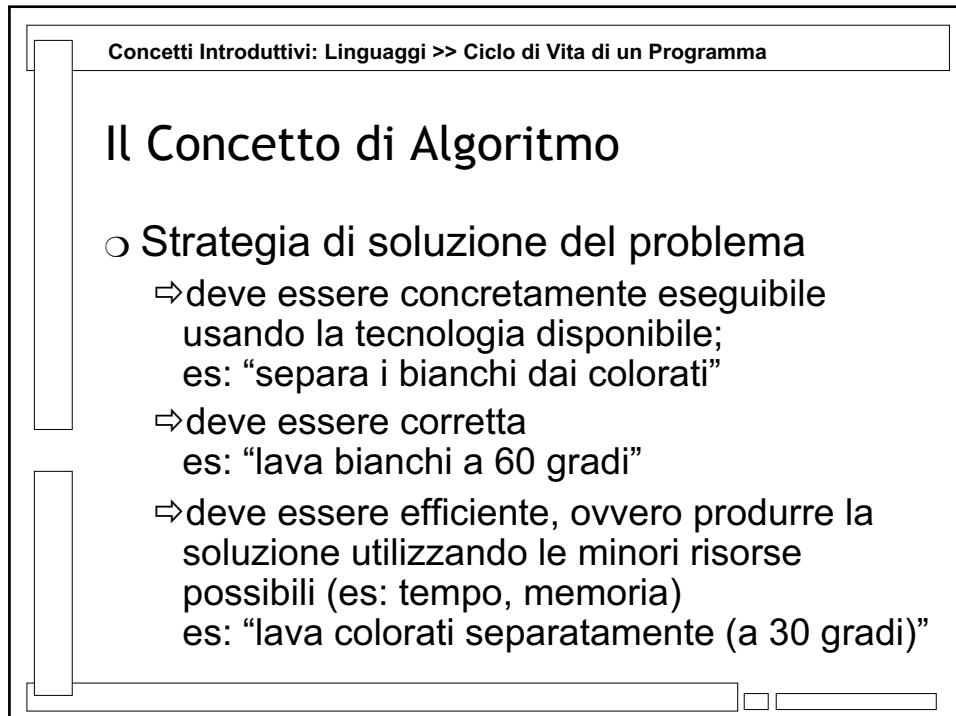
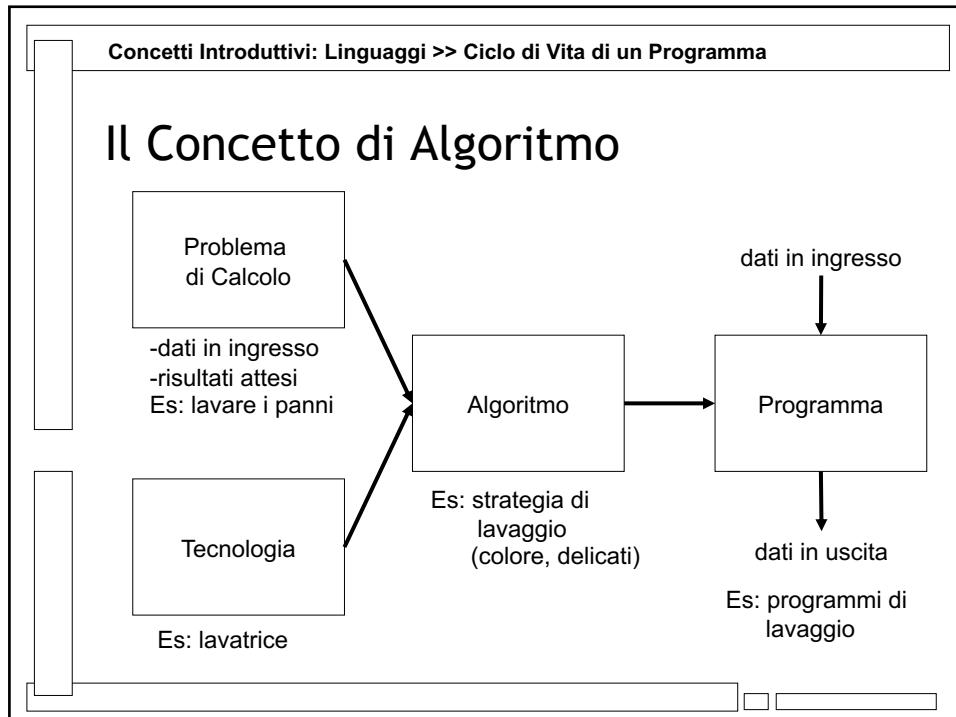
- Per programmare
  - ⇒ è necessario conoscere almeno un linguaggio (sintassi e semantica)
- Ma questo non basta
  - ⇒ il processo di sviluppo del software è complesso
  - ⇒ è necessario un metodo (ovvero una “metodologia”) per affrontare questa complessità

## Il Concetto di Algoritmo

- Il punto di partenza
  - ⇒ la descrizione del problema, normalmente fornita in linguaggio naturale (“specifica”)
  - ⇒ è opportuno analizzarla accuratamente
- Il punto di arrivo
  - ⇒ l'applicazione correttamente funzionante
- Un passo intermedio fondamentale
  - ⇒ concepire una strategia per la soluzione del problema

## Il Concetto di Algoritmo

- Algoritmo
  - ⇒ strategia per la soluzione del problema
- Il problema centrale
  - ⇒ capito il problema, decidere attraverso quale sequenza di passi costruire la soluzione
  - ⇒ è un problema ricorrente, che capita in molti altri contesti
- Esempio: lavare i panni



## Metodologia di Programmazione

- Metodo per affrontare il ciclo di vita
  - ⇒ passi e tecniche per analizzare il problema
  - ⇒ passi e tecniche per scegliere l'algoritmo
  - ⇒ passi e tecniche per scrivere il codice
  - ⇒ passi e tecniche per verificare il codice
  - ⇒ passi e tecniche per manutenere il codice
- La metodologia è importante
  - ⇒ quanto la conoscenza del linguaggio