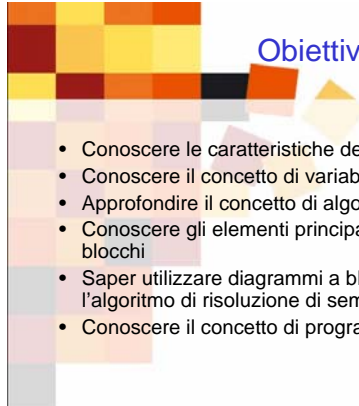




Unità E2

Linguaggi algoritmici

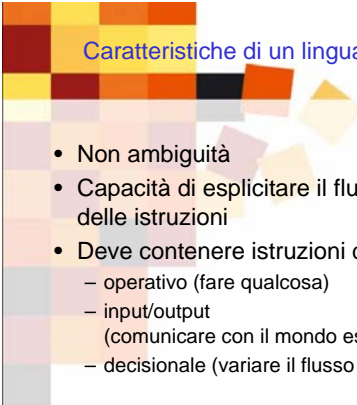
© 2007 SEI-Società Editrice Internazionale, Apogeo



Obiettivi

- Conoscere le caratteristiche dei linguaggi algoritmici
- Conoscere il concetto di variabile e di tipo di dato
- Approfondire il concetto di algoritmo
- Conoscere gli elementi principali dei diagrammi a blocchi
- Saper utilizzare diagrammi a blocchi per realizzare l'algoritmo di risoluzione di semplici problemi
- Conoscere il concetto di programmazione strutturata


© 2007 SEI-Società Editrice Internazionale, Apogeo



Caratteristiche di un linguaggio algoritmico


- Non ambiguità
- Capacità di esplicitare il flusso di esecuzione delle istruzioni
- Deve contenere istruzioni di tipo:
 - operativo (fare qualcosa)
 - input/output (comunicare con il mondo esterno)
 - decisionale (variare il flusso di esecuzione)

© 2007 SEI-Società Editrice Internazionale, Apogeo



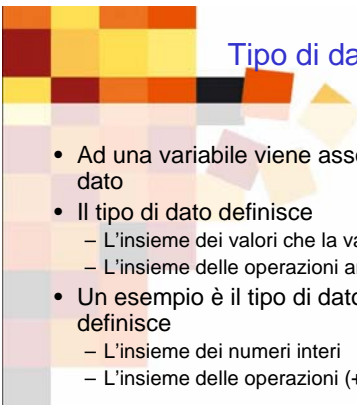
Variabili e memoria

- Una variabile è il nome (identificatore) di un contenitore (zona di memoria) che contiene una informazione (dato)



- L'assegnamento modifica il valore di una variabile
- L'assegnamento è distruttivo (il valore precedente contenuto nella variabile viene perso)

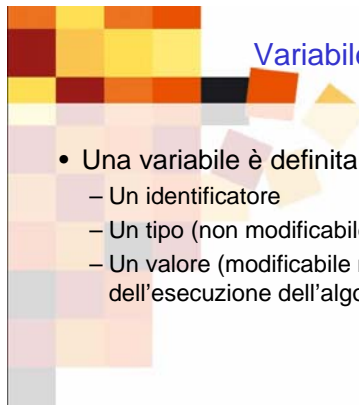
© 2007 SEI-Società Editrice Internazionale, Apogeo



Tipo di dato

- Ad una variabile viene associato un tipo di dato
- Il tipo di dato definisce
 - L'insieme dei valori che la variabile può assumere
 - L'insieme delle operazioni ammesse sui dati
- Un esempio è il tipo di dato intero che definisce
 - L'insieme dei numeri interi
 - L'insieme delle operazioni (+ - * /)

© 2007 SEI-Società Editrice Internazionale, Apogeo



Variabile

- Una variabile è definita da:
 - Un identificatore
 - Un tipo (non modificabile)
 - Un valore (modificabile nel corso dell'esecuzione dell'algoritmo)

© 2007 SEI-Società Editrice Internazionale, Apogeo

Nota: gli identificatori

- E' importante seguire le regole sintattiche definite dal linguaggio per la scelta dei nomi da assegnare alle variabili
- Nei linguaggi di programmazione queste regole saranno rigorose
- E' importante scegliere nomi significativi legati al problema per facilitare la comprensione dell'algoritmo
 - meglio *cateto1*, *cateto2*, *ipotenusa*, *area*
 - di *x*, *y*, *z*, *j*

© 2007 SEI-Società Editrice Internazionale, Apogeo

L'algoritmo e l'esecutore

- L'esecutore deve poter **comprendere** il linguaggio algoritmico
- L'esecutore deve poter **eseguire** le azioni indicate dalle istruzioni dell'algoritmo (istruzioni elementari)

© 2007 SEI-Società Editrice Internazionale, Apogeo

Diagrammi a blocchi

- Linguaggio algoritmico di tipo grafico
- Formato da:
 - **Blocchi** (contenitori di istruzioni)
 - **Frecce** (definiscono il flusso di esecuzione)

© 2007 SEI-Società Editrice Internazionale, Apogeo

Blocco iniziale

- Definisce il punto iniziale dell'algoritmo
- Esiste solo un solo blocco iniziale
- Nessuna freccia in ingresso al blocco
- Una sola freccia in uscita determina il blocco successivo nel flusso di esecuzione

© 2007 SEI-Società Editrice Internazionale, Apogeo

Blocco finale

- L'esecuzione dell'algoritmo termina in questo blocco
- Esiste solo un solo blocco finale
- Nessuna freccia in uscita al blocco (nessuna istruzione successiva)
- Più frecce in ingresso al blocco

© 2007 SEI-Società Editrice Internazionale, Apogeo

Blocchi di input

- L'esecutore riceve dal mondo esterno una informazione (dato) e la inserisce nella variabile
- Una sola freccia in uscita determina il blocco successivo nel flusso di esecuzione
- Più frecce in ingresso al blocco

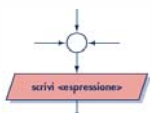
© 2007 SEI-Società Editrice Internazionale, Apogeo

Tipi di input

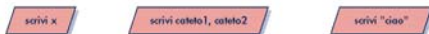
- A seconda del tipo di realtà in esame e del tipo di esecutore l'informazione in input potrebbe essere comunicata all'esecutore in vario modo:
 - Comunicazione verbale
 - Comunicazione scritta
 - Digitazione su una tastiera
 - Strumento di acquisizione
 - Termometro
 - Sensore ottico
 - ...

© 2007 SEI-Società Editrice Internazionale, Apogeo

Blocchi di output



- L'esecutore comunica al mondo esterno una informazione (dato)
- L'informazione può essere il valore di una variabile o un valore costante
- Una sola freccia in uscita determina il blocco successivo nel flusso di esecuzione
- Più frecce in ingresso al blocco



© 2007 SEI-Società Editrice Internazionale, Apogeo

Tipi di output

- A seconda del tipo di realtà in esame e del tipo di esecutore l'informazione in output potrebbe essere dall'esecutore in vario modo:
 - Comunicazione verbale
 - Comunicazione scritta
 - Visualizzazione su uno schermo
 - Segnale elettrico
 - ...

© 2007 SEI-Società Editrice Internazionale, Apogeo

Blocchi operativi



- Una sola freccia in uscita determina il blocco successivo nel flusso di esecuzione
- Più frecce in ingresso al blocco
- Contiene una istruzione imperativa che impone all'esecutore di eseguire una operazione elementare
- L'esecutore deve essere in grado di eseguire l'operazione
- A seconda del tipo di esecutore può essere di vario tipo:
 - Salta
 - Volta a destra
 - Cuoci a fuoco lento per 10 minuti

© 2007 SEI-Società Editrice Internazionale, Apogeo

Tipi di problemi

- In generale tratteremo soprattutto problemi di natura numerica
 - Per la loro semplicità
 - Perché abbiamo mostrato come, attraverso rappresentazioni numeriche digitali (in particolare binarie) è possibile trattare problemi di varia natura
- Nei blocchi operativi troveremo quindi operazioni di assegnamento
 - variabile ← espressione

© 2007 SEI-Società Editrice Internazionale, Apogeo

Notazione per le espressioni

- Per uniformità con il mondo della programmazione introduciamo già da ora una notazione "in linea" per le espressioni
- Utilizzo delle sole parentesi tonde
- Non utilizzo di apici – pedici
- Utilizzo delle funzioni nella forma:
 - nome(argomento)

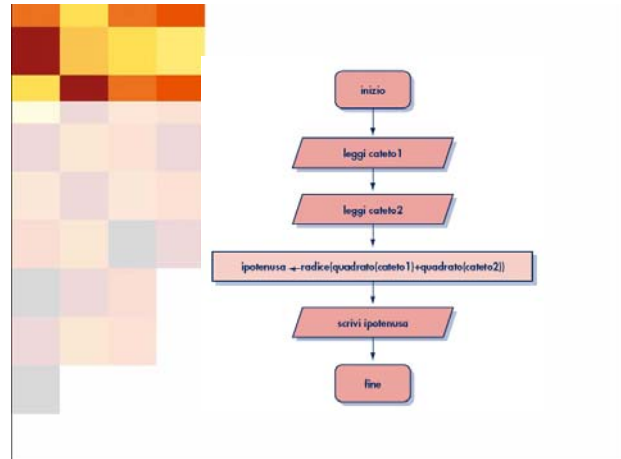
$$\sqrt{\text{Cateto}_1^2 + \text{Cateto}_2^2} \quad \text{radice(quadrato(cateto1) + quadrato(cateto2))}$$

© 2007 SEI-Società Editrice Internazionale, Apogeo

Algoritmi: un esempio numerico

- Problema: date le misure dei cateti trovare la misura dell'ipotenusa in un triangolo rettangolo
- Dati iniziali: misura del cateto1 e del cateto2
- Dato finale: misura dell'ipotenusa
- Esecutore: in grado di comprendere i diagrammi a blocchi e di effettuare le operazioni sui numeri reali compreso elevamento al quadrato – quadrato(x) – calcolo della radice quadrata – radice(x) -

© 2007 SEI-Società Editrice Internazionale, Apogeo



Verifica dell'algoritmo

- L'ultima fase del procedimento di risoluzione di un problema è la verifica
- Si controlla che i dati finali non portino ad una contraddizione con i dati iniziali
- Procedimento: traccia di verifica
- Elenco di tutte le variabili, dell'input e dell'output
- Esecuzione passo per passo dell'algoritmo

© 2007 SEI-Società Editrice Internazionale, Apogeo



Verifica: esempio

| | cateto1 | cateto2 | ipotenusa | input | output |
|--------|---------|---------|-----------|-------|--------|
| passo1 | 40 | - | - | 40 | - |
| passo2 | - | 30 | - | 30 | - |
| passo3 | - | - | 50 | - | - |
| passo4 | - | - | - | - | 50 |

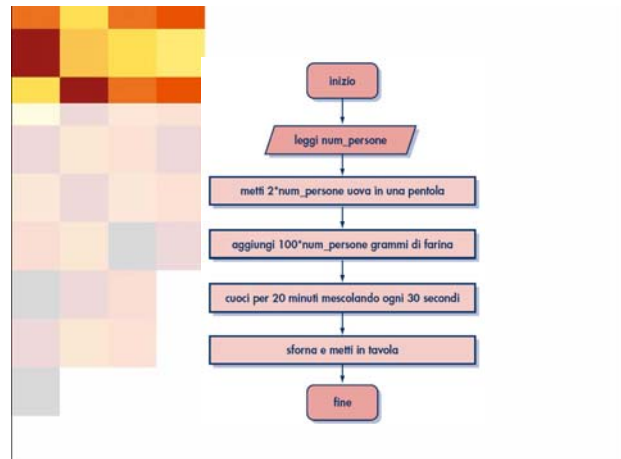
© 2007 SEI-Società Editrice Internazionale, Apogeo



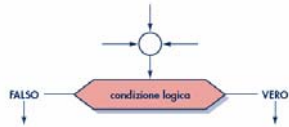
Algoritmi: un esempio non numerico

- Problema: preparare una torta (?!?)
- Dato iniziale: numero di persone
- Dato finale: la torta pronta
- Esecutore: in grado di comprendere i diagrammi a blocchi e di effettuare le operazioni elementari di cucina (mescolare, cuocere ...)

© 2007 SEI-Società Editrice Internazionale, Apogeo



Blocchi decisionali



- Più frecce in ingresso al blocco
- Due frecce in uscita etichettate vero e falso
- Contiene una condizione logica (espressione che può assumere il valore vero o falso)
- Il blocco successivo è definito dal valore della condizione logica
- E' un blocco che altera il flusso di esecuzione dell'algoritmo

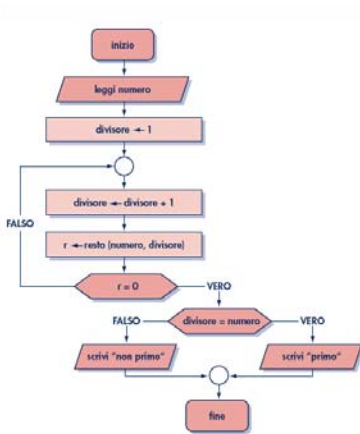


© 2007 SEI-Società Editrice Internazionale, Apogeo

Algoritmi: un esempio numerico

- Problema: verificare se un numero è primo
- Dato iniziale: il numero intero positivo
- Dato finale: "primo" o "non primo"
- Esecutore: in grado di comprendere i diagrammi a blocchi e di effettuare le operazioni matematica compresa il resto della divisione intera $\text{resto}(n,d)$

© 2007 SEI-Società Editrice Internazionale, Apogeo



Verifica: esempio

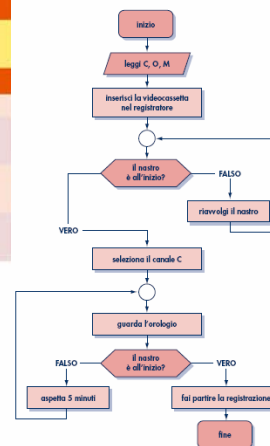
| | numero | divisore | r | input | output |
|--------|--------|----------|---|-------|-----------|
| passo1 | 15 | - | - | 15 | - |
| passo2 | - | 1 | - | - | - |
| passo3 | - | 2 | - | - | - |
| passo4 | - | - | 1 | - | - |
| passo5 | - | 3 | - | - | - |
| passo6 | - | - | 0 | - | - |
| passo7 | - | - | - | - | non primo |

© 2007 SEI-Società Editrice Internazionale, Apogeo

Algoritmi: un esempio non numerico

- Problema: insegnare ad un bambino ad utilizzare un videoregistratore per registrare un film.
- Dati iniziali: Canale, Ora e Minuto di inizio film
- Dato finale: Film registrato su videocassetta
- Esecutore (bambino) è in grado di comprendere i diagrammi a blocchi e di eseguire alcune operazioni elementari (prendere una videocassetta, comprendere il significato dei tasti del videoregistratore ...)

© 2007 SEI-Società Editrice Internazionale, Apogeo



Diagrammi a blocchi: pregi e difetti

- Pregi:
 - Semplicità
 - Possibilità di seguire facilmente il flusso di esecuzione
- Difetti:
 - Per algoritmi complessi si ottiene una struttura molto complessa e risulta difficile decifrare il procedimento seguito nella risoluzione

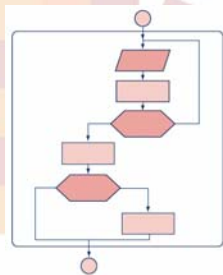
© 2007 SEI-Società Editrice Internazionale, Apogeo

Verso la programmazione strutturata

- La possibilità data dai diagrammi di selezionare l'istruzione successiva senza seguire regole specifiche può portare ad algoritmi difficilmente leggibili e soprattutto difficili da modificare
- Un approccio differente è quello della programmazione strutturata che impone all'insieme delle istruzioni strutture ben definite

© 2007 SEI-Società Editrice Internazionale, Apogeo

La programmazione strutturata



- Basata sul concetto di costrutto strutturato
- Costrutto = insieme di istruzioni con un solo punto d'ingresso ed un solo punto d'uscita

© 2007 SEI-Società Editrice Internazionale, Apogeo

Le strutture fondamentali

- La programmazione strutturata è basata su tre strutture fondamentali:
 - Sequenza
 - Selezione
 - Ripetizione
- Nel 1966 due studiosi italiani (Böhm e Jacopini) hanno dimostrato che un qualsiasi algoritmo può essere espresso usando esclusivamente le tre strutture fondamentali

© 2007 SEI-Società Editrice Internazionale, Apogeo

Algoritmi e programmi strutturati

- I moderni linguaggi di programmazione (prossima unità) seguono tutti le regole della programmazione strutturata
- Per la realizzazioni di algoritmi seguiremo due strade:
 - Diagrammi a blocchi strutturati
 - Pseudolinguaggio algoritmico

© 2007 SEI-Società Editrice Internazionale, Apogeo

Pseudolinguaggio

- Struttura simile a quella dei linguaggi di programmazione
- Non grafico ma "lineare"
- Non sono definiti i dettagli implementativi
- Linguaggio didattico
- La "base" per i linguaggi di programmazione

© 2007 SEI-Società Editrice Internazionale, Apogeo



Sintesi (1)

- I **linguaggi algoritmici** devono essere **non ambigui** e devono poter rappresentare le istruzioni di **input/output**, le istruzioni **operative** e quelle **decisionali** che determinano il flusso di esecuzione.
- Una **variabile** è definita da un **identificatore** e da un **valore** che viene modificato dall'operazione di **assegnamento**.
- Un **tipo di dato** definisce un insieme di **valori** e di **operazioni** effettuabili su questi valori.
- I **diagrammi a blocchi** (flow-chart) sono formati da **blocchi** (inizio/fine, operativi, input/output e decisionali) e da **freccie** che li collegano.

© 2007 SEI-Società Editrice Internazionale, Apogeo



Sintesi(2)

- Tutti i blocchi possono avere **più frecce in ingresso** e **una sola freccia in uscita**, tranne i blocchi decisionali che hanno due uscite etichettate vero e falso.
- Tramite la **traccia di verifica** è possibile controllare la correttezza del procedimento algoritmico.
- Per evitare diagrammi complessi da comprendere, viene introdotto il concetto di **costrutto sintattico strutturato**.
- Oltre ai diagrammi a blocchi esistono altri linguaggi algoritmici come lo **pseudolinguaggio** o **linguaggio di progetto**, che ha caratteristiche simili ai linguaggi di programmazione ma non deve essere eseguito dall'elaboratore.

© 2007 SEI-Società Editrice Internazionale, Apogeo

