Calcolo numerico e **programmazione**
Programmazione

Tullio Facchinetti
<tullio.facchinetti@unipv.it>

11 maggio 2012

14:05

http://robot.unipv.it/toolleeo

La programmazione

la programmazione è l'insieme delle attività che il programmatore svolge per creare un programma

il processore si occupa di eseguire le istruzioni del programma

Il programma

il programma consiste nella sequenza delle azioni, il cosiddetto algoritmo, che devono essere eseguite per realizzare il compito desiderato

il termine programma è spesso usato erroneamente in modo intercambiabile con altri termini, come software o applicazione (una applicazione può essere composta da diversi programmi)

Il programma

le operazioni elementari per il funzionamento di un programma sono 4:

1. trasferimento di informazioni: acquisizione dati, visualizzazione risultati intermedi, scrittura risultati finali
2. esecuzione di calcoli
3. assunzione di decisioni: scelta della successiva operazione da compiere sulla base di risultati intermedi
4. esecuzione di iterazioni: ripetizione di sequenze di operazioni

Il programma

per descrivere un algoritmo non è possibile utilizzare il linguaggio naturale che può presentare ambiguità che potrebbero causare interpretazioni false o errate

si utilizzano linguaggi sintetici e standardizzati in modo da consentire all'esecutore una interpretazione univoca

Esecuzione del programma

è la fase con la quale le istruzioni rappresentate in linguaggio macchina vengono messe in esecuzione dal processore

le tipiche operazioni compiute sono

- caricamento in memoria, tipicamente a partire da una periferica di memoria di massa, come un disco rigido
- identificazione del "punto d'ingresso" del programma
- esecuzione sequenziale delle istruzioni (fetch + esecuzione)

Errori e debug

l'errore di programmazione viene universalmente chiamato bug

esistono errori di tipo sintattico, semantico e logico

"fare il debug" di un programma significa ricercare e correggere gli errori

Tipologie di errori

- errori di sintassi
- errori semantici
- errori logici

Correttezza di un programma

l'area di un triamgolo e base per altezza

- non è sintatticamente corretta
- contiene una parola che non è di senso compiuto

Correttezza di un programma

l'area di un triangolo e base per altezza

- corretta sintatticamente: non contiene parole o costrutti non validi
- è composta da due frasi di senso compiuto unite dalla congiunzione "e"
- non è corretta dal punto di vista semantico: sostanzialmente non significa nulla

Correttezza di un programma

una frase corretta sia sintatticamente che semanticamente:

> l'area di un triangolo è base per altezza

- la frase è corretta sia sintatticamente che semanticamente
- non è corretta dal punto di vista logico: in questo caso l'affermazione è falsa

Correttezza di un programma

una frase corretta dal punto di vista sintattico, semantico e logico è la seguente:

> l'area di un rettangolo è base per altezza

Errori di sintassi

- sono relativamente semplici da trovare
- infrangono delle regole ben definite per la scrittura del codice
- sono segnalati in modo automatico dagli strumenti usati per lo sviluppo di un programma

es.

- scrivere wile invece di while
- dimenticare di chiudere una parentesi precedentemente aperta

Errori semantici

- sono in genere segnalati in modo automatico
- sono ricercati su programmi sintatticamente corretti

es.

- richiamare una funzione che non esiste
- confrontare un numero intero con una stringa
- assegnare un valore ad una costante (3 = x)
- assegnare un valore ad una espressione (x + y = 3)

Errori logici

- sono più difficili da identificare
- sono collegati alla logica di funzionamento del programma
- è difficile rilevarli per mezzo di procedure automatiche
- si manifestano tipicamente in fase di esecuzione del programma, cosa che complica ulteriormente il debugging
- spesso dipendono dai dati in ingresso

es.

- effettuare un ciclo per un numero di volte errato
- combinare in modo errato più test nelle istruzioni condizionali

Testing

viene effettuato su un programma sintatticamente e semanticamente corretto (quindi eseguibile)

> il programma viene collaudato per verificarne la correttezza logica

si verifica che l'algoritmo implementato svolga correttamente le operazioni previste

> si realizza fornendo in ingresso al programma opportuni valori di input per verificare che l'output corrispondente sia corretto

Manutenzione

spesso il programmatore non deve sviluppare ex novo un programma, ma si trova a dover modificare un programma

è un aspetto talvolta trascurato del ciclo di vita di un programma

- spesso il programma è stato scritto da altri programmatori, oppure
- è stato scritto dallo stesso programmatore, ma
- è passato un periodo di tempo sufficiente da far dimenticare i dettagli dell'implementazione

1. utilizzare uno stile di programmazione chiaro e coerente
2. commentare il codice

Linguaggi di programmazione

il programma viene realizzato scrivendo del codice sorgente utilizzando un linguaggio di programmazione

esistono molti diversi linguaggi di programmazione, ciascuno dei quali ha caratteristiche specifiche che lo rendono adatto a compiti specifici

il linguaggio adatto per l'applicazione specifica!

Classificazione dei linguaggi

- linguaggi interpretati vs compilati
- linguaggi di basso livello vs alto livello
- linguaggi procedurali
- linguaggi funzionali
- linguaggi dichiarativi
- linguaggi ad oggetti
- linguaggi di scripting

le tipologie di linguaggi di programmazione
non sono esclusive
(es. linguaggio compilato, di alto livello, funzionale)

Linguaggi di basso vs alto livello

per livello di un linguaggio si intende la sua vicinanza al modo di rappresentare il codice rispetto alla macchina che deve eseguirlo piuttosto che al programmatore che deve scriverlo

linguaggi di basso livello

- linguaggi vicini alla rappresentazione usata dalla macchina
- tipicamente più complicato da scrivere e/o comprendere

linguaggi di alto livello

- linguaggi vicini alla rappresentazione umana
- tipicamente più "descrittivi" e facili da scrivere e/o leggere

Linguaggi interpretati vs compilati

differenzia lo stadio al quale viene analizzato il codice sorgente

linguaggi interpretati

- il codice sorgente viene interpretato ed eseguito direttamente da un apposito programma chiamato interprete
- generalmente più lenti in quanto c'è l'overhead dovuto all'esecuzione dell'interprete

linguaggi compilati

- richiedono che il codice sorgente, una volta terminato, sia processato da un compilatore che lo converte in linguaggio macchina e ne permette l'esecuzione da parte della CPU

Linguaggi interpretati vs compilati

linguaggi interpretati

- es. BASIC, Perl, Python, MATLAB

linguaggi compilati

- es. C, Pascal

Linguaggi procedurali

l'organizzazione di un programma è basato su blocchi logico-funzionali chiamati procedure o funzioni

- la funzione raggruppa un insieme di istruzioni e/o chiamate ad altre funzioni che implementano funzionalità specifiche e ben definite
- la divisione di un programma in funzioni rende più chiara la stesura del codice
- una stessa funzione può essere richiamata varie volte nel corso del programma

es. C, ...

Linguaggi dichiarativi

non viene implementato direttamente un algoritmo

- il programmatore non specifica come deve essere ottenuto il risultato, ovvero non implementa un algoritmo
- il programmatore indica quali sono i dati coinvolti nel calcolo del risultato e qual è il risultato desiderato
- le azioni per mettere in relazione i dati al fine di ottenere il risultato desiderato sono individuate e compiute da un interprete

es. Prolog, make

Linguaggi a oggetti

evoluzione della programmazione procedurale
con l'introduzione degli oggetti

gli oggetti sono caratterizzati da:

1. incapsulamento, cioè l'oggetto incorpora sia i dati che le funzioni che operano sui dati
2. ereditarietà
3. polimorfismo

es. C++, Java

Linguaggi di scripting

servono ad automatizzare l'esecuzione di lunghe attività sequenziali

- sono linguaggi nati per descrivere una sequenza di esecuzione di altri programmi o comandi (esecuzione batch)
- in seguito sono state aggiunte funzionalità quali l'esecuzione di cicli e l'uso di variabili
- sono tutti linguaggi interpretati

es. PHP, Perl, JavaScript, Python, shell Unix