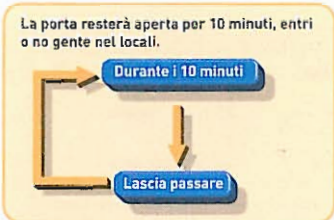


Struttura di programmazione

Dopo aver visto i programmi realizzati sino a questo momento abbiamo già un'idea chiara di che cosa sia una struttura di programmazione, o struttura di controllo. Ciò che abbiamo trattato negli altri capitoli era incentrato sul LetPicBasic; ora però lo faremo in termini più generali, includendo anche ciò che possiamo trovare negli altri linguaggi di programmazione.

Definizione di struttura di programmazione

Le strutture di controllo sono elementi che non possono mancare in nessun linguaggio di programmazione, dato che permettono di eseguire nel programma diverse azioni, oppure di agire o non agire sotto determinate condizioni. Quindi i



I microrobot hanno bisogno dell'intelligenza fornita dalle strutture di controllo.

programmi non sono lineari né si eseguono sempre nello stesso modo. Questo fatto, pur se importante in qualsiasi programma, quando si lavora con i microrobot se possibile riveste un'importanza ancora maggiore, dato che i microrobot basano la loro intelligenza sui dati che ricevono dall'esterno e su come agire in base a questi dati, e questo non sarebbe possibile senza le strutture di controllo.

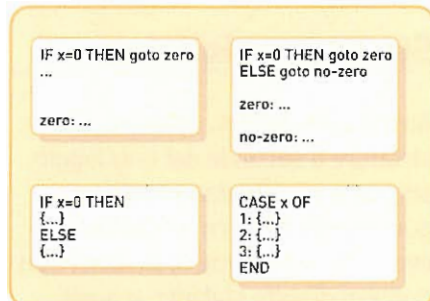
Tipi di strutture di controllo

Le strutture di programmazione si possono dividere principalmente in due gruppi: le prese di decisione o strutture condizionali, e i cicli. Con le prime lo sviluppo del programma può prendere un percorso oppure un altro a seconda di diverse condizioni con le quali si "decide" fra le diverse opzioni, da cui deriva il nome. I cicli permettono di realizzare lo stesso processo molte volte senza dover riscrivere la stessa linea, o linee, di codice più volte. Sono anche conosciute come strutture cicliche. Anche se questi due gruppi sono piuttosto comuni in tutti i linguaggi di programmazione, esistono altri due tipi di strutture che in alcuni linguaggi appaiono come gruppi differenziati. Sono le strutture di gestione degli errori e i comandi di uscita. Data l'importanza delle strutture di controllo in qualsiasi linguaggio di programmazione,

vedremo ora una breve descrizione per distinguerle, e in capitoli successivi le tratteremo poi nel dettaglio.

Strutture per la presa di decisione

Abbiamo visto che lavorando con il LetPicBasic disponiamo solo di una struttura fra quelle chiamate di presa di decisione: la condizionale IF...THEN. In questa istruzione a seconda del valore di una espressione le cui due possibili risposte possono essere vero o falso {1 o 0 logico} si realizza un'azione. In questo caso



Differenti tipi di istruzioni condizionali.

qualunque sia il risultato dell'espressione si prosegue con l'istruzione seguente, a meno che l'istruzione da compiere sia un'istruzione "goto". Questa istruzione, pur se proibita in molti linguaggi di programmazione dato che può causare una destrutturazione del programma, è necessaria in alcuni casi per realizzare salti all'interno del



programma che in un altro modo sarebbe impossibile fare. Di per sé non è una struttura decisionale né tanto meno una struttura ciclica, però aiuta entrambe nei salti. In altri linguaggi esistono anche le strutture IF...THEN...ELSE, dove si prende un percorso oppure un altro a seconda se l'espressione valutata sia vera o falsa: si tratta di una struttura a due rami. In alcuni linguaggi però potremo trovare strutture a rami multipli: come "switch" o "case", in cui le espressioni non solo danno come risultato vero o falso, ma anche diversi valori come numeri con cui si comparano per eseguire una istruzione oppure l'altra.

Nella presa di decisione potrebbe essere sufficiente una sola istruzione per verificare il compimento della condizione, oppure un insieme di istruzioni, questo dipenderà dal linguaggio.

Strutture cicliche

Anche qui troviamo differenti strutture a seconda del linguaggio, anche se i tipi fondamentali sono due: quelli che permettono di eseguire un insieme di istruzioni un numero di volte stabilito, e quelli che le eseguono "mentre" oppure "sino a che" si compie una condizione. In tutti i casi se non si dispone di queste strutture si possono costruire partendo dalle strutture condizionali e dalle istruzioni "goto". Nel LetPicBasic possiamo contare su FOR...NEXT, che realizza le istruzioni il numero di volte che noi indichiamo. Le istruzioni di chiamata a subroutine o funzioni, interrompono anch'esse il flusso normale del programma, anche se non sono strutture di controllo propriamente

dette. In altri linguaggi possiamo anche trovare strutture cicliche del secondo tipo, ognuna di esse pensata per un'interazione distinta, anche se è sempre possibile convertire l'una nell'altra. Ad esempio, WHILE, ripete una istruzione un numero "n" di volte sconosciuto, mentre si compie una condizione, però data la sua struttura può succedere che non esegua nulla perché l'espressione non si compie o non si entra nel ciclo. Invece DO...WHILE esegue l'insieme di istruzioni almeno una volta e dopo valuta l'espressione, quindi se vogliamo che si comporti come quella precedente dobbiamo variare il range di valori

```
x=0
WHILE x<1 DO
begin
x=x+1
...
end

x=0
DO
x=x+1
...
UNTIL x=1
```

Lo stesso ciclo si può esprimere in diversi modi.

possibili di questa espressione. DO...UNTIL eseguirà le espressioni "sino a che" si compie la condizione, anche se si può adattare per fare in modo che il suo funzionamento sia come quello delle precedenti. Un'altra procedura piuttosto comune è inserire un ciclo all'interno di un altro, cosa che è conosciuta come "cicli annidati". Serve per fare alcune elaborazioni più complesse di quelle viste sinora. Bisogna inoltre tener conto che il numero di interazioni di ogni ciclo si moltiplica per quello dei cicli annidati, per ottenere il valore totale delle sequenze.

```
FOR x=1 TO 10 DO
FOR y=1 TO 20 DO
FOR z=1 TO 30 DO
```

Le istruzioni verranno eseguite 10*20*30 volte

Nel caso di cicli annidati le interazioni si moltiplicano.

Altre strutture di controllo

Alcuni linguaggi di programmazione dispongono di strutture speciali che permettono di conoscere se si è generato un errore durante l'esecuzione del codice, sono le strutture per la gestione degli errori. In ultimo i comandi di uscita, come indica il loro nome, sono utilizzati per uscire da un'interazione. Non sono, quindi, strutture di controllo ma come le istruzioni "goto" aiutano queste strutture. Tre di questi comandi sono BREAK, che si utilizza per fermare un ciclo tornando immediatamente al flusso del programma; CONTINUE, che esce dall'interazione però per iniziarne un'altra nuova all'interno dello stesso ciclo, cioè torna all'inizio del ciclo, e RETURN che torna alla chiamata a subroutine o funzione.



Differenze fra due comandi di uscita.

