Attività di laboratorio n.5

scheda allegata 1

Proposta di lavoro

L'attività ha lo scopo di familiarizzare gli alunni all'uso di un linguaggio di programmazione, alle strutture fondamentali dell'informatica, alla comprensione e all'uso del concetto di variabile.

L'attività è articolata in 8 lezioni della durata di un'ora.

Lezione 1
Il docente invita gli alunni a scrivere la sequenza di istruzioni utile a far disegnare alla Tartaruga un quadrato di lato 100. Dopo qualche discussione si arriva a definire la sequenza operativa, che viene trascritta sul computer collegato alla LIM, nella riga di comando:

A 100
D 90
A 100
D 90
A 100
D 90
A 100

Il docente clicca su REIMPOSTA, cancellando ciò che la Tartaruga ha disegnato sullo schermo, e invita un alunno a far disegnare ancora alla Tartaruga un quadrato. Di fronte al motivato mugugno della classe, spiega che questa difficoltà di dover riscrivere tutte le volte la sequenza operativa necessaria a risolvere un problema può essere superata insegnando alla Tartaruga nuove parole (nuove istruzioni che si aggiungano alle istruzioni primitive conosciute dalla Tartaruga).
Per far ciò, bisogna scrivere **edita "quadrato** nella riga di comando.
Si apre una nuova finestra in cui si trovano già scritte le parole

Per quadrato
Fine

Si scrive la sequenza operativa precedente fra **Per quadrato** e **Fine**, e si clicca infine su FILE>SALVA ED ESCI.
Se ora si digita **quadrato** nella riga di comando, la Tartaruga esegue il compito, e cliccare su REIMPOSTA non rappresenta più un guaio perché basta digitare ancora **quadrato** per far eseguire nuovamente il compito alla Tartaruga.
Abbiamo così definito una nuova istruzione per la Tartaruga, o per meglio dire abbiamo scritto una PROCEDURA, che è già in tutto e per tutto un programma.

A questo punto il docente esce deliberatamente dall'ambiente MSWLOGO e, dopo essere rientrato nell'ambiente, chiede ad un alunno di cercare di far eseguire la procedura **quadrato**. L'alunno scrive **quadrato**, ma la Tartaruga risponde con un messaggio d'errore: "non so come fare **quadrato**".
Cos'è successo?
Il docente propone questa metafora: *quando si scrive **edita "quadrato** la Tartaruga mette a disposizione un foglietto su cui viene scritta la PROCEDURA. Quando si clicca su FILE>SALVA ED ESCI, la Tartaruga inserisce il foglietto in una sua tasca detta TASCARAM, che è molto capiente. Quando la PROCEDURA viene **invocata**, la Tartaruga (che non conosce l'istruzione **quadrato**) guarda nella sua TASCARAM, e se trova un foglietto chiamato **quadrato** esegue le istruzioni primitive che vi sono contenute. Dopo si rimette il foglietto in tasca. Quando si esce dall'ambiente MSWLOGO è come spegnere la Tartaruga/computer, per cui la Tartaruga prima di spegnersi prende tutti i foglietti contenuti nella sua TASCARAM e li butta via (la RAM è una memoria volatile!). Sarà quindi opportuno dire alla Tartaruga di conservare i foglietti con la/le procedura/e in un cassetto esterno (una cartella contenuta nell'hard-disk), attraverso l'opzione FILE>SALVA CON NOME.*
Quando si inizia una nuova sessione, sarà sufficiente cliccare su FILE>APRI perché la Tartaruga si rimetta nella TASCARAM tutti i foglietti che abbiamo scritto in precedenza, e sia pronta ad eseguirli.

Il docente fa notare che se aggiungiamo un D 90 alla sequenza:

A 100
D 90
A 100
D 90
A 100
D 90
A 100

possiamo individuare un modulo che si ripete 4 volte: A 100 D 90.
Questa aggiunta rende TRASPARENTE la PROCEDURA (una procedura è trasparente quando alla fine di essa la Tartaruga si trova nella stessa posizione che aveva all'inizio) e consente di utilizzare l'istruzione primitiva RIPETI, la cui sintassi è: RIPETI n [azione1 azione2 ….].

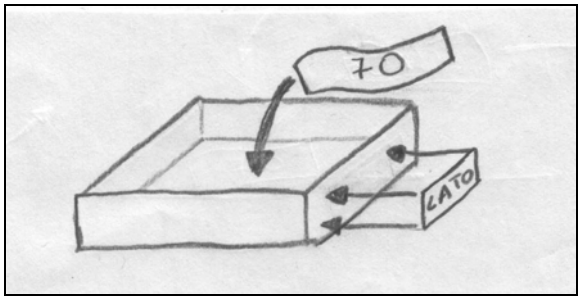
La procedura **quadrato** diventa allora:

Per quadrato
ripeti 4 [A 100 D 90]
Fine

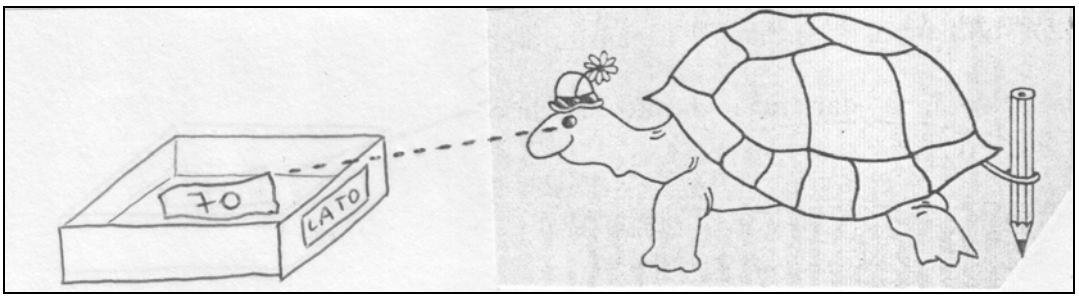
Lezione 2
Dopo aver caricato l'archivio contenente la procedura **quadrato**, il docente propone un problema: cosa bisogna fare se vogliamo far disegnare alla Tartaruga un quadrato di lato 50? Evidentemente sarà necessario scrivere un'altra procedura (che però non potrà chiamarsi **quadrato**, ma ad esempio **quad**) in cui l'**argomento** di avanti sia 50.
Ognuna di queste procedure risolve solo un singolo problema (in altre parole disegna un singolo quadrato). Sono cioè sequenze operative e non algoritmi. Gli algoritmi (lo ricordiamo) risolvono intere classi di problemi.
Per poter far disegnare alla Tartaruga tutti i quadrati possibili (e questo è già un piccolo algoritmo) dobbiamo rendere **variabile** l'argomento dell'istruzione avanti.
Dovremo quindi, nel momento della **definizione** della procedura, scrivere:

Per quadrato :lato
ripeti 4 [A :lato D 90]
Fine

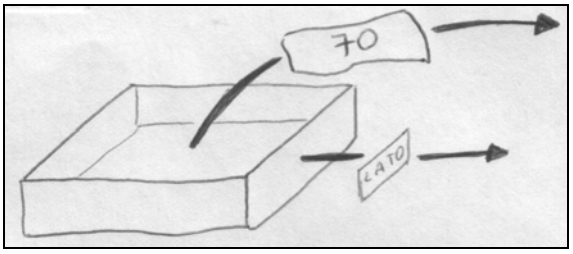
Cosa fa la Tartaruga quando viene invocata la procedura **quadrato**?
Il docente propone questa metafora: *innanzitutto, al momento dell'**invocazione** la parola **quadrato** dovrà essere seguita da un numero, ad esempio 100 oppure 50 oppure 35, altrimenti la Tartaruga mostrerà un messaggio d'errore: "argomenti insufficienti per quadrato", esattamente come succede con le primitive avanti e sinistra.*
*Se abbiamo scritto **quadrato 70**, la Tartaruga prima di iniziare ad eseguire la procedura prepara una scatoletta vuota (ne ha tantissime a disposizione nella sua TASCARAM) e, dato che dopo i due punti c'è la parola LATO, incolla un'etichetta con la scritta LATO sulla scatola e vi inserisce un foglietto su cui è scritto il numero 70. Quindi inizia ad eseguire la procedura.*



*Ogni volta che dopo l'istruzione **avanti** trova **:lato**, va a vedere quale valore è contenuto nella scatoletta LATO e, verificato che è 70, avanza di 70 passi.*



*Questo si ripete 4 volte e, giunta alla parola **Fine**, la Tartaruga getta via il foglietto contenuto nella scatola di nome LATO e toglie l'etichetta dalla scatola.*



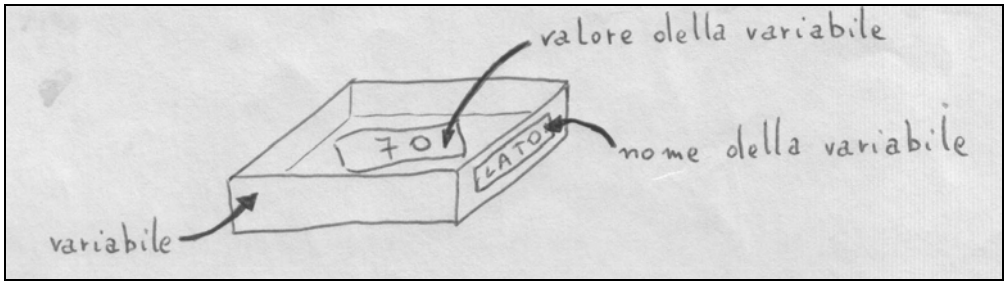
È possibile verificare l'annullamento della variabile digitando nella riga di comando

stampa :lato

La Tartaruga risponderà che

lato non ha valore

Stampa è un'istruzione primitiva che riporta il valore attuale contenuto in una variabile. In questo caso dunque la scatola di nome LATO e il valore che viene inserito dentro di essa sono validi solo all'interno della procedura. Diremo che la scatola (la variabile) di nome LATO è una scatola (una variabile) privata della procedura **quadrato**. L'argomento di un'istruzione può quindi essere rappresentato da una scatoletta a cui possiamo **assegnare** diversi valori. Questa scatoletta viene chiamata **variabile**. La parola con cui etichettiamo la scatoletta si chiama **nome della variabile** mentre il numero che inseriamo nella scatoletta si chiama **valore della variabile**.



Il docente propone allora di scrivere, sulla falsariga della procedura **quadrato**, una procedura **triaequil** che disegni triangoli equilateri.

Dopo aver ovviamente modificato l'argomento di ripeti da 4 a 3, gli alunni provano sui loro computer a modificare l'argomento di destra, inserendo vari valori e verificando se viene disegnato un triangolo equilatero. Alla fine si scopre che l'istruzione giusta è D 120.
Il docente chiede se gli alunni conoscono il valore dell'angolo interno di un triangolo equilatero. Si conviene che è di 60 gradi, e che questo valore si ottiene attraverso la regola per cui **la somma degli angoli interni di qualsiasi triangolo è sempre di 180 gradi** e, dato che **gli angoli di un triangolo equilatero sono uguali fra di loro**, ognuno di essi vale 180 : 3= 60
Si scopre così che, contrariamente a quanto gli alunni avevano immaginato nel quadrato (dove sibillinamente l'angolo interno e l'angolo esterno si equivalgono), **la rotazione della Tartaruga corrisponde all'angolo esterno supplementare all'angolo interno** (un angolo è supplementare ad un altro se, sommandoli, il loro valore complessivo risulta di 180 gradi).

La procedura **triaequil** risulterà quindi scritta così:

```
Per triaequil :lato
ripeti 3 [A :lato D 120]
Fine
```

Il docente assegna come compito a casa l'individuazione della procedura per disegnare pentagoni regolari.

Lezione 3
Nessun alunno ha risolto il problema e il docente riconosce che in effetti è difficile scoprire di quanto deve ruotare la Tartaruga per disegnare un pentagono. In realtà c'è una regola che dice che **la somma degli angoli interni di un poligono si ottiene moltiplicando il valore dell'angolo piatto per il numero dei lati del poligono meno due**.
[SommaAi = 180 * (N-2)]
E allora
Ai = SommaAi : N

Ma proviamo a risolvere il problema in modo sperimentale. Per disegnare il quadrato la Tartaruga ha ruotato di 90 gradi per 4 volte (cioè tante volte quanti sono i lati), per un totale di 360 gradi. Per disegnare il triangolo ha ruotato di 120 gradi per 3 volte, per un totale di 360 gradi. Vuoi vedere che per disegnare il pentagono regolare dovrà ruotare 5 volte per un totale di 360 gradi? Se così fosse, il valore della rotazione (angolo esterno supplementare all'angolo interno) si otterrebbe facilmente eseguendo l'operazione 360 : 5 = 72.

Gli alunni scrivono la procedura

```
Per pentreg :lato
ripeti 5 [A :lato D 72]
Fine
```

e verificano che effettivamente viene disegnato un pentagono regolare. Provano ad applicare la scoperta anche all'esagono e funziona ancora.

```
Per esareg :lato
Ripeti 6 [A :lato D 60]
Fine
```

Si giunge così a definire una regola generale, chiamata **il teorema del giro completo della Tartaruga**: **per disegnare una qualsiasi figura piana, la somma delle rotazioni della Tartaruga deve essere uguale a 360**. Quindi, in un poligono regolare, basterà dividere 360 per il numero dei lati per ottenere il valore della rotazione della Tartaruga.

Il docente suggerisce che, invece di sobbarcarci noi la fatica di fare le divisioni, possiamo approfittare del fatto che la Tartaruga è un computer, o come si diceva una volta un calcolatore. Quindi perché non far fare a lei i calcoli necessari? Li farà senz'altro più velocemente e più esattamente.
La procedura per realizzare un poligono di 15 lati diventerà allora:

```
Per poliquindici :lato
ripeti 15 [A :lato D 360/15]
Fine
```

Gli alunni provano a scrivere sulla falsariga di questa procedura altre procedure per creare tanti poligoni regolari diversi, scoprendo tra l'altro che man mano che il numero dei lati aumenta il poligono somiglia sempre di più ad un

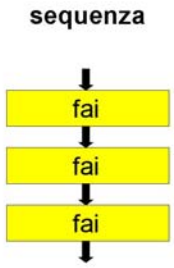
cerchio, e sperimentando quindi direttamente la definizione del **cerchio** come un **poligono regolare con un numero infinito di lati**.

Il docente chiede di osservare le procedure scritte, invitando a notare una loro particolarità: l'argomento di ripeti è uguale al divisore dell'argomento di destra, e tutti e due questi valori corrispondono al numero dei lati del poligono. Come scrivere allora una procedura che disegni tutti i poligoni regolari, qualsiasi sia il valore del lato e qualsiasi sia il numero dei lati? Sarebbe un algoritmo molto potente.
Diventa semplice ed intuitivo ripetere l'esperienza della variabile del lato per definire una variabile che rappresenti il numero dei lati.

```
Per poliregolari :lato :numerolati
ripeti :numerolati [A :lato D 360/:numerolati]
Fine
```

Il docente invita a ripensare cosa fa la Tartaruga di fronte all'invocazione **poliregolari 100 5**, piuttosto che all'invocazione **poliregolari 70 13** (le scatole delle variabili) e invita gli alunni a sperimentare con valori diversi.

Lezione 4
Il docente ricorda le tre strutture fondamentali dell'informatica: sequenza, selezione e ripetizione.
In LOGO la struttura della sequenza è semplice.



Sequenza in LOGO

```
A 100
D 90
A 100
D 90
A 100
D 90
```

Prima di esaminare l'uso della selezione e della ripetizione nel LOGO, il docente introduce una particolare caratteristica di questo linguaggio, comune però anche ad altri linguaggi di programmazione: la **ricorsione**.
Chiede quindi agli alunni se conoscono una nota filastrocca:

C'era una volta un re
Seduto sul sofà
Che chiese alla sua serva:
"Raccontami una storia"
La serva incominciò:
C'era una volta un re
Seduto sul sofà
Che chiese alla sua serva:
"Raccontami una storia"
La serva incominciò:
eccetera……

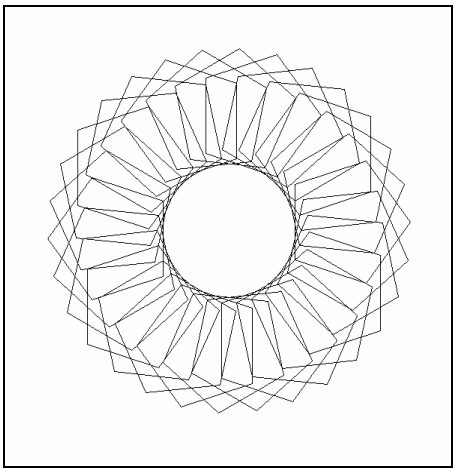
La ricorsione consiste, in questo caso, in una filastrocca che richiama se stessa. Analogamente in LOGO la ricorsione consiste in una procedura che richiama se stessa. Con gli alunni si conviene che il procedimento tende all'infinito senza mai fermarsi.
Prima di sperimentare sui computer l'uso della ricorsione in LOGO, si cercano altri esempi conosciuti di procedimenti ricorsivi (mettersi in mezzo a due specchi, un libro il cui titolo sia la domanda: "Qual è il titolo di questo libro?").
Il docente suggerisce di provare a scrivere la seguente procedura:

```
Per giocoinfinito
A 1
giocoinfinito
Fine
```

Gli alunni verificano facilmente che la procedura non si ferma più, eseguendola la Tartaruga (appunto) all'infinito.
Il docente invita gli alunni ad inserire una chiamata ricorsiva nella procedura **poliregolari**, e quindi ad inserire prima della chiamata ricorsiva qualche istruzione che sposti la Tartaruga dal punto iniziale (la CASA della Tartaruga).
Così facendo, si disegnano facilmente forme regolari che tendono a ricoprire tutto lo schermo senza che la Tartaruga si fermi mai.
Ad esempio:

```
Per poliregolari :lato :numerolati
ripeti :numerolati[A :lato D 360/:numerolati]
SU A 10 D 34 A 11 S 46 GIU
poliregolari :lato :numerolati
Fine
```

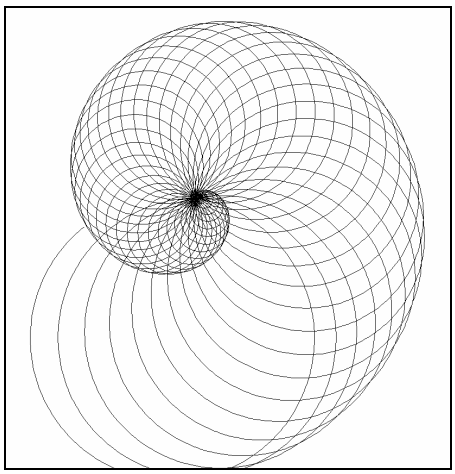
Invocando poliregolari 100 5, si otterrà il seguente disegno:



Il docente spiega che incrementando il valore della variabile LATO di 5 nella chiamata ricorsiva, e inserendo D 10 prima della stessa chiamata ricorsiva sarà possibile realizzare delle spirali.

```
Per poliregolari :lato :numerolati
ripeti :numerolati[A :lato D 360/:numerolati]
D 10
poliregolari :lato+5 :numerolati
Fine
```

Invocando **poliregolari 5 7**, si disegna una spirale a modulo ettagonale



Che però ricopre rapidamente lo schermo, non fermandosi mai.

Lezione 5
Come fare a fermare l'esecuzione della procedura ricorsiva da parte della Tartaruga?
Qui ci viene in aiuto la struttura della selezione.

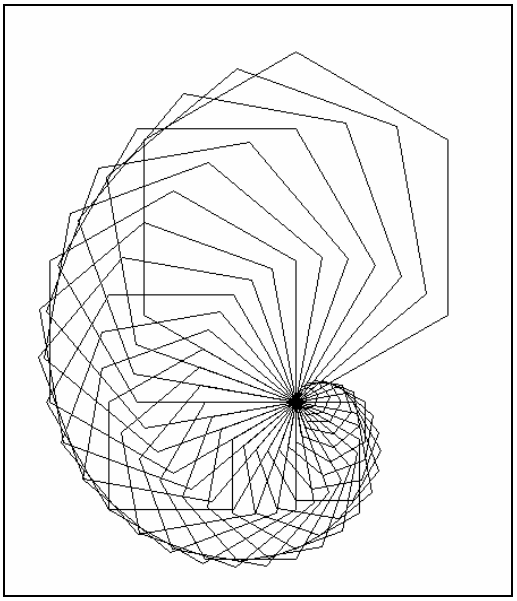
Selezione in LOGO

SE condizione [azione1 azione2]

Bisogna considerare che nella procedura **poliregolari :lato :numerolati**, il valore della variabile LATO si incrementa di 5 ad ogni chiamata ricorsiva. Se all'inizio vale 5, dopo 30 chiamate ricorsive il suo valore sarà di 155. Possiamo dire quindi alla Tartaruga che SE il valore di LATO supera 150, ALLORA deve fermarsi.
In LOGO scriveremo così:

```
Per poliregolari :lato :numerolati
ripeti :numerolati[A :lato D 360/:numerolati]
D 10
SE :lato>150 [STOP]
poliregolari :lato+5 :numerolati
Fine
```

Invocando **poliregolari 5 6**, otterremo questo disegno, fermando la Tartaruga dopo un certo numero di chiamate ricorsive.



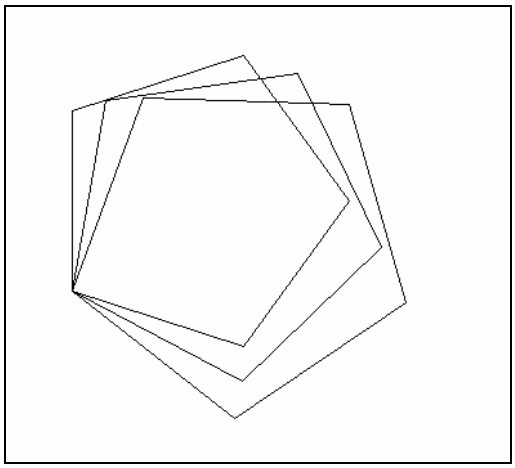
Gli alunni sperimentano la procedura utilizzando valori diversi per le variabili LATO e NUMEROLATI.

Lezione 6
Il docente suggerisce di rendere più ordinata la scrittura delle procedure, distinguendo **poliregolari** dalla procedura **spirale**.

```
Per poliregolari :lato :numerolati
ripeti :numerolati[a :lato d 360/:numerolati]
Fine
```

```
Per spirale :lato :numerolati
poliregolari :lato :numerolati
d 10
se :lato>150[stop]
spirale :lato+10 :numerolati
Fine
```

Il controllo viene effettuato dalla Tartaruga sul valore della variabile LATO.
Se però, al momento dell'invocazione, diamo a LATO un valore vicino a 150 (ad esempio spirale 140 5), otterremo una spirale con pochissimi pentagoni.



Come possiamo fare per ottenere comunque una spirale con un certo numero di poligoni, indipendentemente dal valore del LATO?
Il docente introduce il concetto di **contatore**, che funziona più o meno come il contatore della corrente elettrica o come il contachilometri dell'auto.
Con l'istruzione primitiva **assegna** viene innanzitutto inizializzata la variabile "c assegnandole valore 0. Per far ciò sarà opportuno definire una nuova procedura **spirale1**, in modo che l'inizializzazione non sia coinvolta nella ricorsione.

```
Per spirale1 :lato :numerolati
assegna "c 0
spirale :lato :numerolati
Fine
```

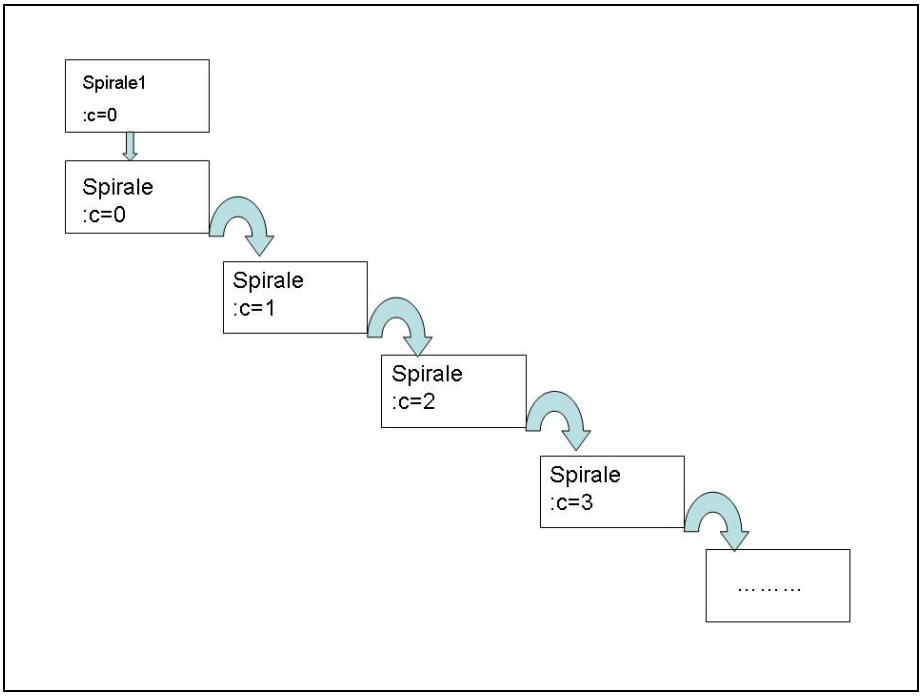
Poi, si inserisce un'altra istruzione **assegna** che **incrementa il contatore** di 1 ad ogni chiamata ricorsiva. Nella struttura di controllo che ferma l'esecuzione della procedura dovremo quindi inserire la variabile C (**controllo del contatore**).

```
Per spirale :lato :numerolati
poliregolari :lato :numerolati
d 10
assegna "c :c+1
se :c>35[stop]
spirale :lato+10 :numerolati
Fine
```

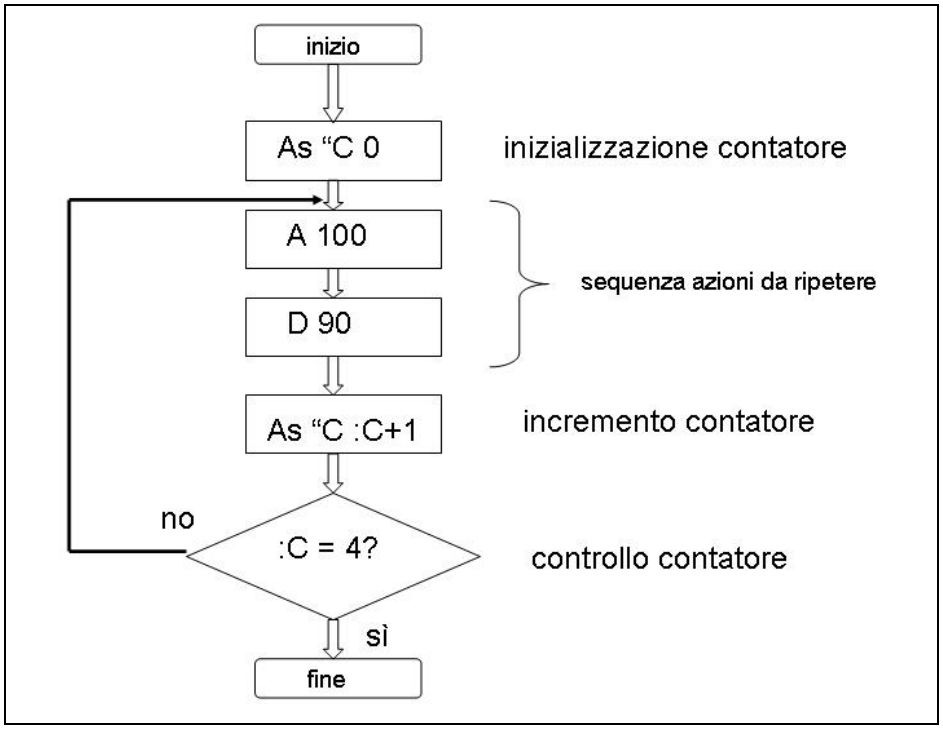
Questa modifica consente di ottenere spirali di 36 poligoni qualsiasi sia il valore iniziale del LATO.

Il docente sottolinea questo secondo modo di **assegnare** un valore ad una variabile, diverso da quello usato all'inizio e che consisteva nel far seguire un numero al nome della procedura nel momento dell'invocazione (es. spirale 5 8).
L'istruzione primitiva **assegna** permette di assegnare un valore ad una variabile durante l'esecuzione di una procedura. Ciò rende **pubblica** la variabile (nell'altro modo essa resta privata); questo significa che la Tartaruga non getterà via il foglietto contenuto nella scatola "C , che resterà a disposizione di ogni altra procedura fino all'uscita dall'ambiente LOGO. Se infatti una volta eseguita la procedura **spirale1** si digita nella riga di comando STAMPA :C , la Tartaruga comunicherà il valore attuale della variabile "C, scrivendo il numero 36. Questo spiega perché è necessario scrivere la riga di inizializzazione a 0 della variabile nella procedura **spirale1**.
Il docente evidenzia anche la differenza fra "C (nome della variabile) e :C (valore della variabile).

Lezione 7
Il docente schematizza graficamente alla LIM la procedura ricorsiva per realizzare spirali, ricordando che essa andrebbe avanti all'infinito se la condizione :c>35 non ne stoppasse bruscamente l'esecuzione.



Introdotto il concetto di contatore possiamo finalmente esaminare il funzionamento della struttura di ripetizione, la terza struttura fondamentale dell'informatica, che nel LOGO viene eseguita con l'istruzione primitiva RIPETI.
Cosa succede, ad esempio, quando viene eseguita l'istruzione RIPETI 4[A 100 D 90] ?
Per farlo capire, il docente schematizza graficamente alla LIM il diagramma di flusso della struttura della ripetizione, utilizzando il contatore e sottolineando la differenza anche grafica con la ricorsione; quest'ultima infatti non è rappresentabile con un diagramma di flusso, non possiede (essendo infinita) una proprietà fondamentale dell'algoritmo: la finitezza.



Lezione 8
Somministrazione del test di verifica