

Kturtle

primi passi con la tartaruga

25 febbraio 2008

3 marzo

10 marzo

Marinella Tomai

KTurtle

E' un software che serve per programmare, il linguaggio che si utilizza e Logo. La prima versione di Logo è stata creata da Papert negli anni '70 nel laboratorio sull'intelligenza artificiale del MIT: Il matematico Papert è uno dei pionieri dell'intelligenza artificiale. Egli è internazionalmente riconosciuto come grande pensatore sui modi di cambiare l'apprendimento grazie ai computer. Nato e formatosi in Sudafrica, dove ha partecipato attivamente al movimento anti-apartheid, Papert ha svolto ricerca matematica all'Università di Cambridge dal 1954 al '58. Ha collaborato in seguito con Jean Piaget all'Università di Ginevra dal 1958 al '63. E' stata questa esperienza che lo ha portato a considerare l'utilizzo della matematica al servizio della comprensione di come i bambini possano apprendere e pensare. Agli inizi degli anni '60 Papert è entrato al MIT (Massachusetts Institute of Technology), dove ha fondato con Marvin Minsky il Laboratorio di Intelligenza Artificiale (Artificial Intelligence Laboratory). I

Per accedere a Kturtle:

Applicazioni

Educazione

KTurtle

Nella parte superiore dello schermo compare **la barra dei menù**, la parte sotto è divisa in due parti, a sinistra **l'editor del codice**, a destra **l'area di disegno**.

Nell'**editor del codice** si inseriscono i comandi Logo, usando le modalità che si usano in un qualsiasi programma di scrittura, i **comandi**, però, vanno scritti *uno per riga*.

In caso di errore si apre una *Finestra di dialogo d'errore* nella quale vi sono le indicazioni relative all'errore e alla sua posizione (in quale riga si trova). Per questo motivo è utile lavorare nell'editor del codice con i numeri di riga. Per farli comparire, dalla barra del menù:

Visualizza

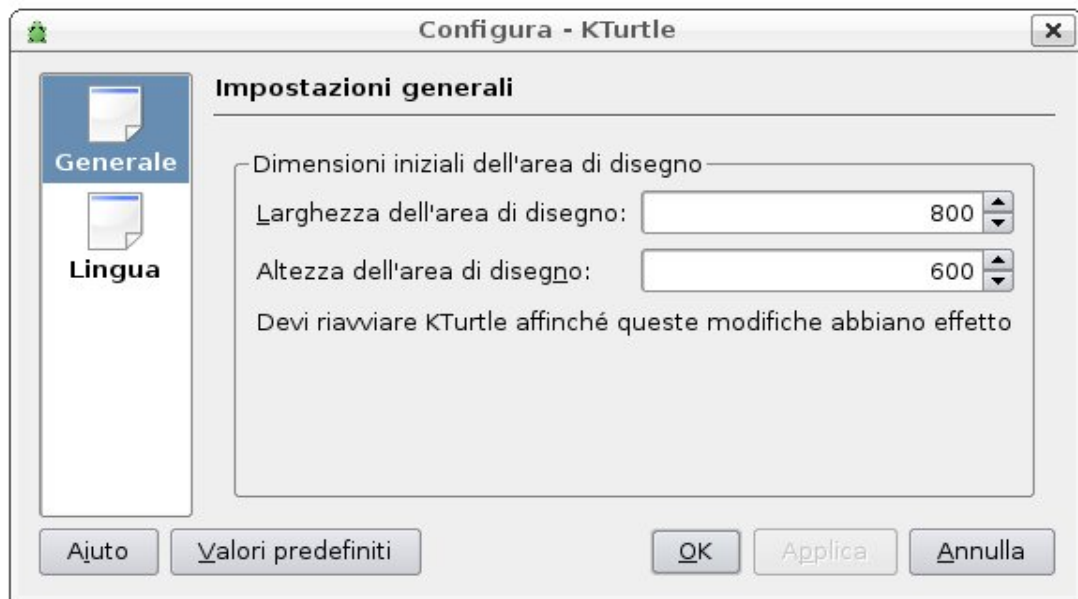
Mostra numeri di riga.

L'area di disegno è la parte dello schermo in cui si muove e disegna la tartaruga, che normalmente si trova al centro dello schermo.

L'area di disegno può avere dimensioni differenti, generalmente si cerca di avere un'area grande. Per modificare le dimensioni:

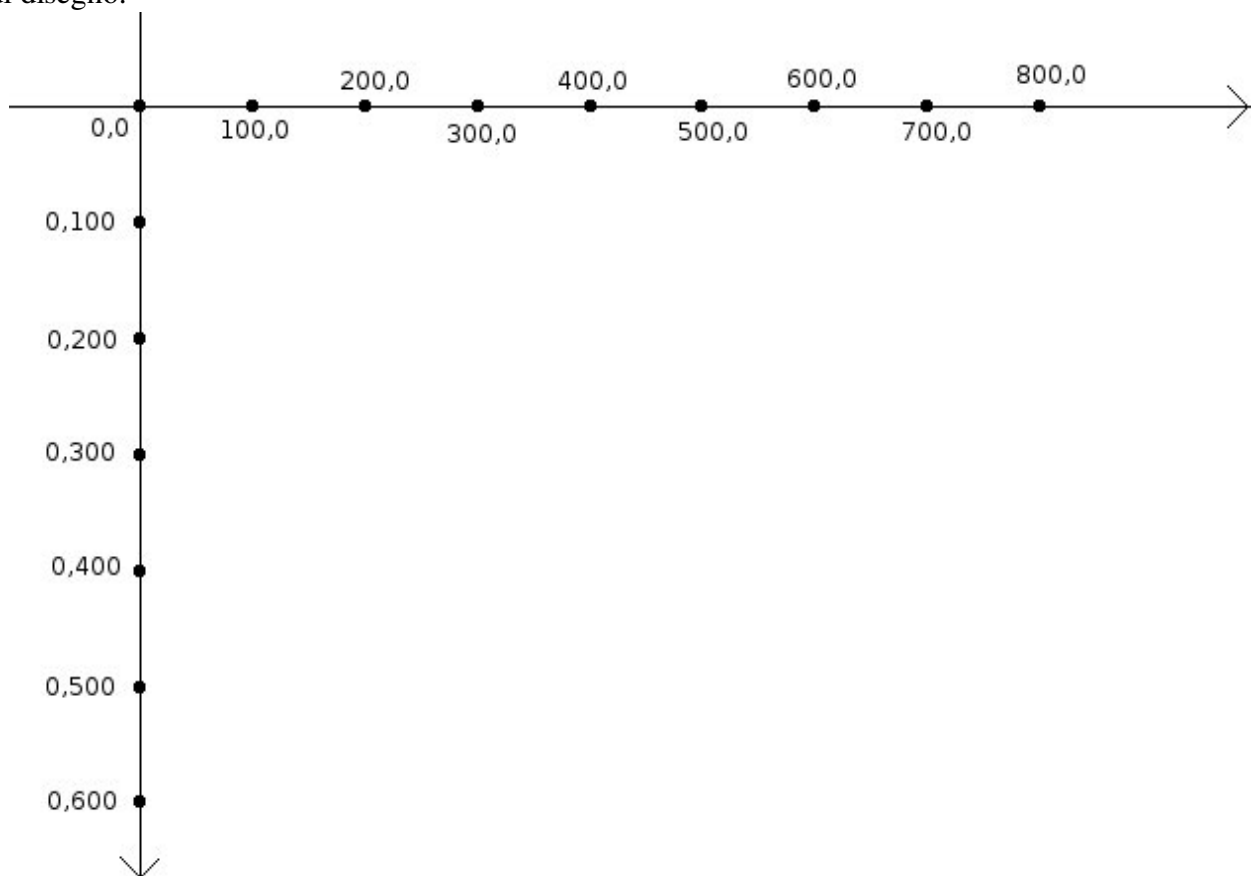
Impostazioni

Configura Kturtle



Se la lingua dei comandi non fosse italiano cliccare, in configura KTurtle, Lingua italiano (it). Dopo aver modificato le impostazioni generali bisogna chiudere il software e avviarlo nuovamente.

L'area di disegno rappresenta un piano cartesiano che ha l'origine nel vertice in alto a sinistra dell'area di disegno.



E' importante tenere presente che l'area di disegno è un piano cartesiano, perché con il comando **vai x,y**, (scrivendo al posto di x un valore compreso tra 0 e 800 e per y un valore compreso tra 0 e 600) si può posizionare la tartaruga in un punto qualsiasi dello schermo. La tartaruga va nel punto di coordinate x,y senza lasciare traccia sullo schermo.

Comandi principali per far muovere la tartaruga nell'area di disegno

- av n** (avanti n) la tartaruga va avanti di n passi
- in n** (indietro n) la tartaruga va indietro di n passi
- dx n** (destra n) la tartaruga ruota in senso orario di n gradi
- sx n** (sinistra n) la tartaruga ruota in senso antiorario di n gradi

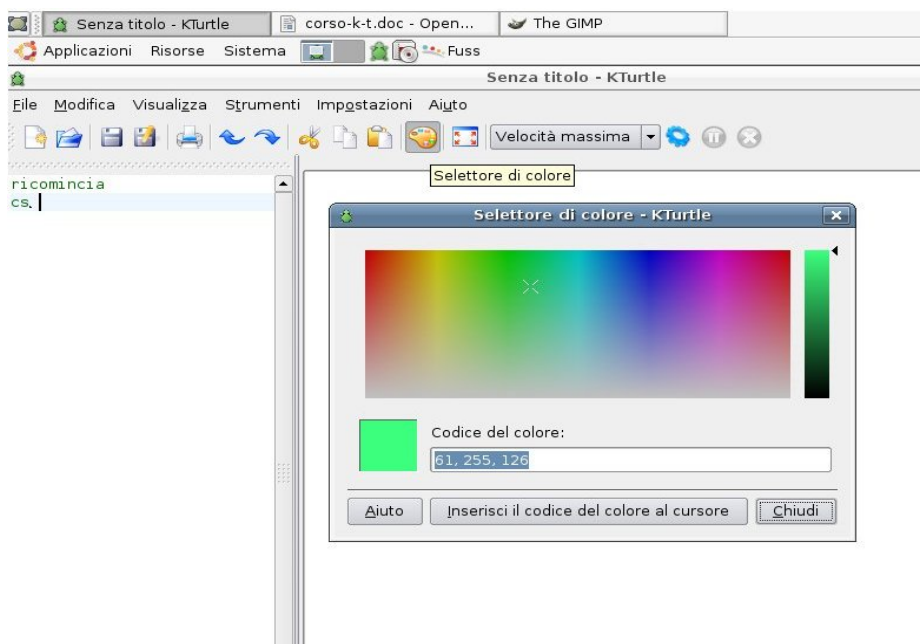
ricomincia dopo questo comando la tartaruga ritorna sempre al centro dello schermo con la “testa” rivolta verso l'alto. E' molto utile usare ricomincia come primo comando di qualunque procedura. Siamo così sicuri che la tartaruga cancelli tutto quello che c'è sullo schermo e si posizioni al centro, in caso contrario la tartaruga rimane nella posizione e nella direzione in cui si trovava precedentemente.

Comandi per modificare l'aspetto grafico (colore, dimensione tratto...)

cs n, n, n **coloresfondo**: imposta il colore dello sfondo

cp n, n, n **colorepenna**: imposta il colore della penna

i numeri che vengono inseriti dopo cs e cp sono la combinazione dei colori RGB che si ottengono dal selettore di colore (icona tavolozza colori nel menù principale)



sp n spessorepenna: imposta lo spessore della traccia lasciata dalla tartaruga. Lo spessore del tratto dipende dal parametro **n** (numero di pixel) che si assegna a sp e che può variare da uno alla dimensione massima dello schermo. Alcuni alunni hanno trovato il modo di colorare disegni variando lo spessore della penna.

Procedure

Nel linguaggio Logo una procedura è un insieme di comandi che mi permettono di realizzare un “prodotto” finale.

Per esempio, se voglio che la tartaruga mi disegni un rettangolo, dovrò scrivere nell'editor del codice una sequenza di comandi:

ricomincia av 300 dx 90 av 200 dx 90
av 300 dx 90 av 200 dx 90

Se poi clicco l'icona del menù principale **esegui i comandi** (rotellina blu), la tartaruga disegnerà sullo schermo un rettangolo.

Ripeti n

Osservando la sequenza delle istruzioni si può vedere che gli stessi comandi si ripetono due volte, quindi, la nostra procedura può essere abbreviata:

ricomincia ripeti 2 [av 300 dx 90 av 200 dx 90]
--

I comandi che si ripetono vanno racchiusi tra parentesi quadrate.

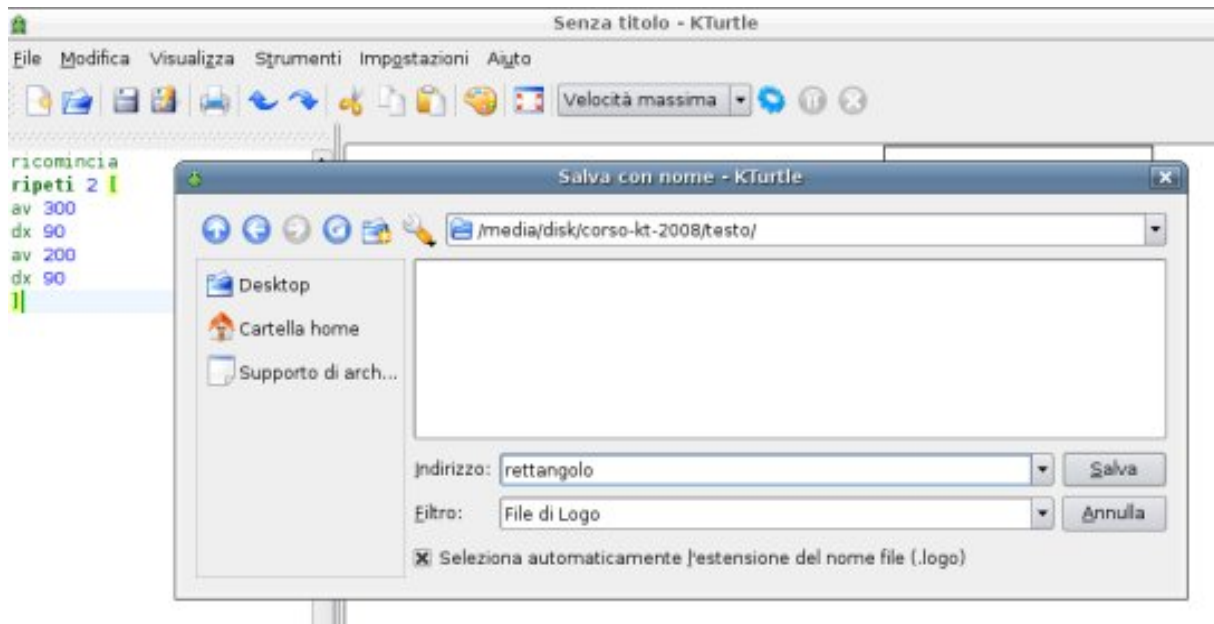
La procedura può essere salvata su un file al quale, ovviamente, si attribuisce un nome.

Per salvare, dal menù principale:

File

Salva con nome

rettangolo



La procedura che mi permette di disegnare il rettangolo è però limitata perché mi disegna sempre rettangoli uguali. Si possono, per costruire rettangoli diversi, utilizzare le *variabili*.

Una variabile può essere rappresentata o da una *lettera* o da una *parola alla quale si può attribuire un qualsiasi valore numerico*.

Per esempio il *file rettangolo* viene così modificato

ricomincia

a = 100

b = 200

ripeti 2 [

av a

dx 90

av b

dx 90

]

Quando si vogliono modificare le dimensioni del rettangolo basta attribuire ad *a* e *b* valori diversi.

Impara

Quando una sequenza di comandi (procedura) deve essere usata più volte all'interno della realizzazione di un progetto, si può attribuirle un nome. Supponiamo che il nostro progetto richieda di disegnare più rettangoli, si può allora dare alla procedura rettangolo il nome "rettangolo", se ci sono delle variabili, queste vanno scritte subito dopo il nome, separate da una virgola. Esempio:

ricomincia

impara rettangolo a,b [

ripeti 2 [

```
av a
dx 90
av b
dx 90
]
]
```

Se si clicca nel menù principale sull'icona **esegui comandi** la tartaruga non farà nulla, perché compaia il rettangolo si deve scrivere:

```
rettangolo 200,300
```

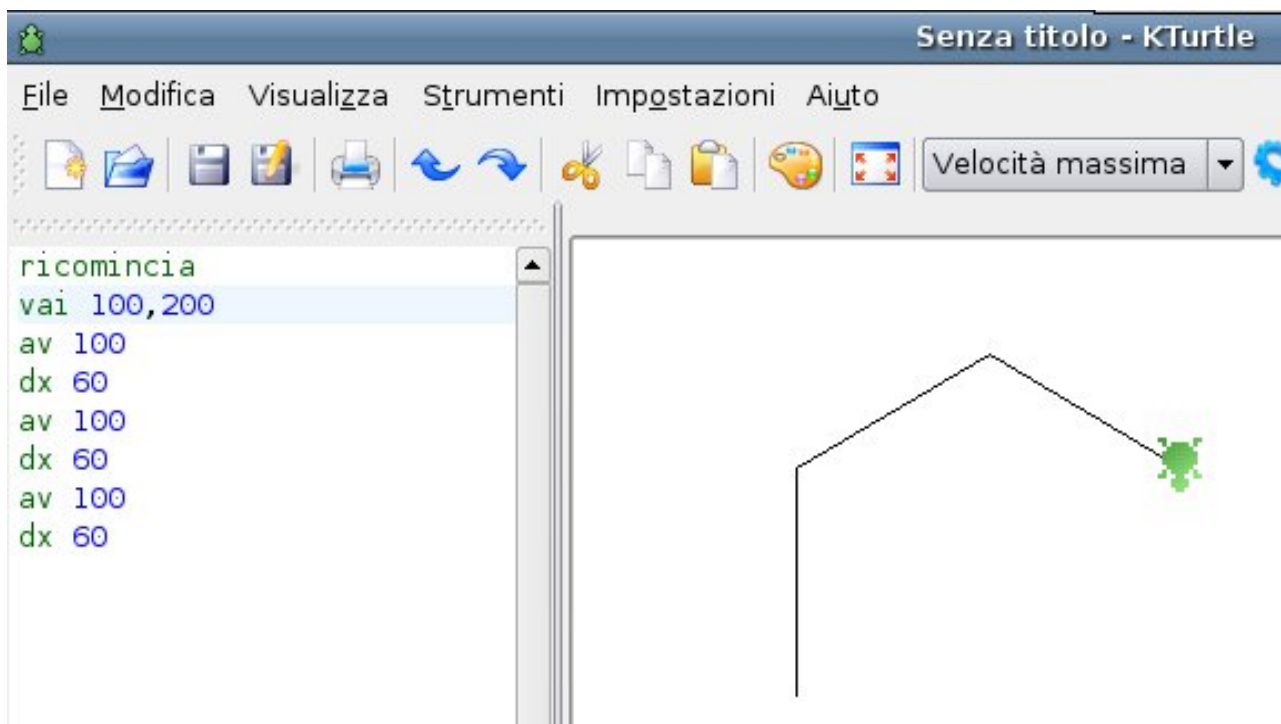
La tartaruga disegnerà un rettangolo con **a** di 200 pixel e **b** di 300 pixel.

Quando gli angoli non sono retti

Le prime difficoltà gli alunni le incontrano quando gli angoli delle figure che vogliono rappresentare non sono retti.

Tipico è l'errore che fanno quando, dopo aver creato le procedure del rettangolo e del quadrato, devono disegnare un triangolo equilatero.

Si riflette con loro sulla figura, osservano che ha tutti gli angoli di 60° (la somma degli angoli interni è 180° e gli angoli sono uguali). La procedura che la maggior parte scrive è



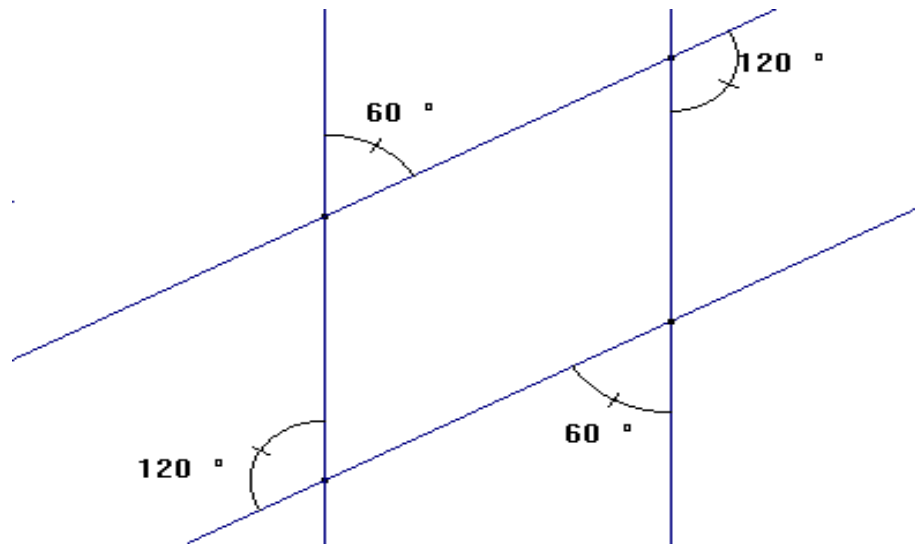
Per realizzare una procedura gli alunni devono rendersi conto che **la rotazione è sempre data dall'angolo** esterno, che nel caso del triangolo equilatero è 120° .

E' indispensabile quindi che gli alunni partano dall'analisi di una figura disegnata sul quaderno, immaginando di essere loro stessi una tartaruga che lascia una traccia.

Procedura per disegnare un parallelogramma

La procedura per disegnare un parallelogramma, tenendo conto degli angoli esterni, sarà

```
Ricomincia  
ripeti 2 [  
av 100  
dx 60  
av 150  
dx 120  
]
```



Se si vogliono usare le variabili non solo per i lati, ma anche per gli angoli bisogna notare che le ampiezze di due angoli consecutivi sono sempre legate tra loro in quanto gli angoli consecutivi sono anche angoli supplementari, quindi si introdurranno tre variabili $a = 100$ (misura di un lato), $b = 150$ (misura di un lato), $n = 60$ (ampiezza dell'angolo)

```
Ricomincia  
a= 100  
b = 150  
n = 60  
ripeti 2 [  
av a  
dx n  
av b  
dx 180-n  
]
```

Se si vuole usare il comando *impara*

```
Ricomincia  
impara para a,b,n [  
ripeti 2 [  
av a  
dx n  
av b  
dx 180-n  
]  
]
```

Procedura per disegnare un poligono regolare

Anche per realizzare una procedura generale valida per la rappresentazione di tutti i poligoni regolari si deve tenere presente che la rotazione è data dall'angolo esterno del poligono.

Per calcolare l'ampiezza dell'angolo esterno si può procedere in differenti modi.

Si può partire da un poligono regolare inscritto in una circonferenza (in questo caso un pentagono) osservando che il poligono si può dividere in cinque triangoli isosceli ($AO = OB$ perché raggi) che hanno l'angolo al vertice

$$AOB = 360^\circ : 5 = 72^\circ$$

$$OAB + ABO = 180^\circ - 72^\circ = 108^\circ$$

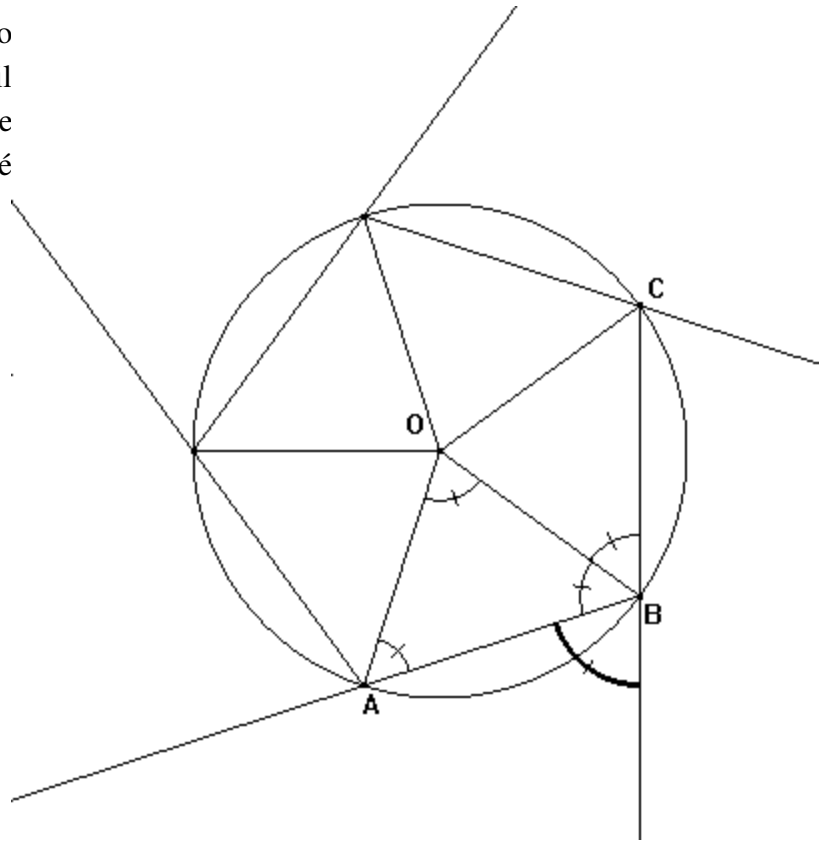
$$OAB = ABO = 108^\circ : 2 = 54^\circ$$

L'angolo interno del pentagono sarà

$$ABC = ABO \times 2 = 54^\circ \times 2 = 108^\circ$$

e l'angolo esterno

$$ABM = 180^\circ - 108^\circ = 72^\circ$$



La dimostrazione si può fare per qualunque poligono regolare.

Un modo più facile può essere, dopo aver rappresentato su un foglio un poligono regolare (più semplici l'esagono o l'ottagono), disegnare gli angoli esterni, ritagliarli e porli uno vicino all'altro per verificare la proprietà della somma degli angoli esterni che è di 360° . Poiché il poligono è regolare (ha gli angoli interni uguali), avrà anche gli angoli esterni uguali e per calcolare l'ampiezza di ciascun angolo basterà dividere 360° per il numero degli angoli, e quindi dei lati, del poligono.

La **rotazione** che dovrà quindi fare la tartaruga **per disegnare un qualsiasi poligono di n lati sarà $360^\circ:n$.**

La procedura sarà

ricomincia

ripeti n [

av l

dx $360/n$

]

dove la variabile **n** rappresenta il **numero dei lati del poligono** e **l** la **lunghezza dei lati**.

Il comando casuale

E' un comando che permette di far scegliere al computer il valore da assegnare ad una certa variabile. Il computer stabilirà *casualmente* quale valore attribuire ad una variabile, tra un minimo ed un massimo da noi scelti.

Ad esempio il comando `vai x,y` fa spostare la tartaruga sullo schermo in un punto di coordinate x,y.

Per far spostare la tartaruga in un punto a caso sullo schermo basterà scrivere:

ricomincia

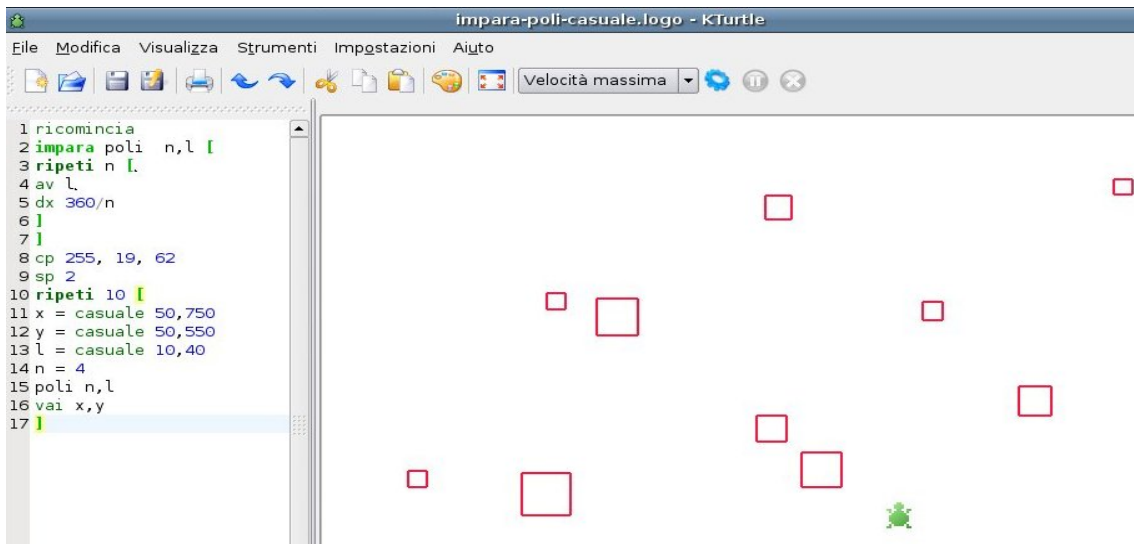
`x = casuale 50, 750`

`y = casuale 50, 650`

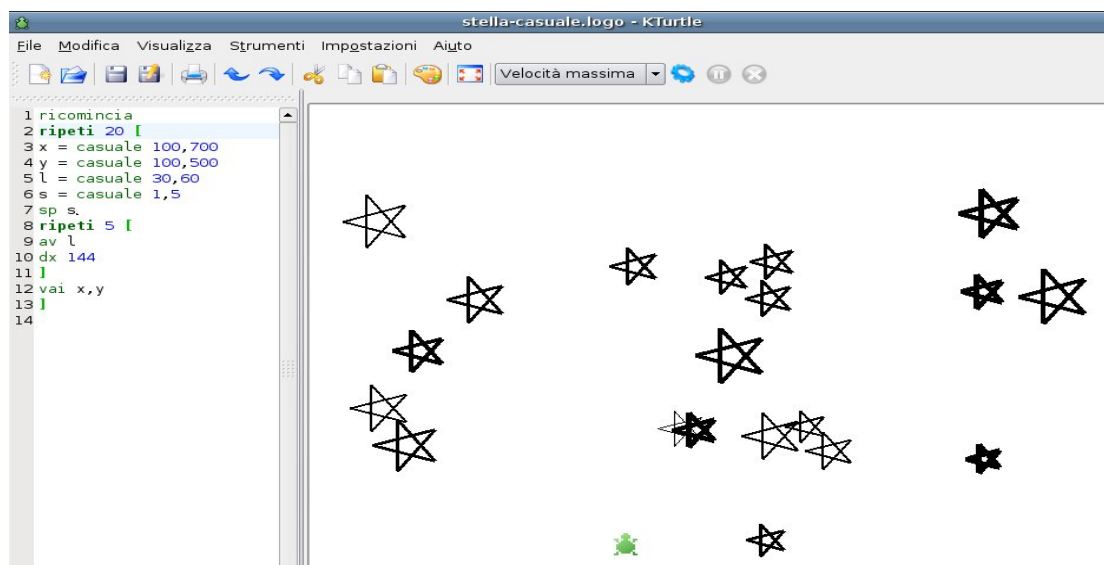
`vai x,y`

Se eseguiremo più volte la procedura, la tartaruga si posizionerà sempre in punti differenti dello schermo.

Sempre con il comando casuale si possono rappresentare poligoni con i lati di dimensioni differenti (variabile lato casuale) e in differenti posizioni dell'area di disegno (`vai x,y casuale`).



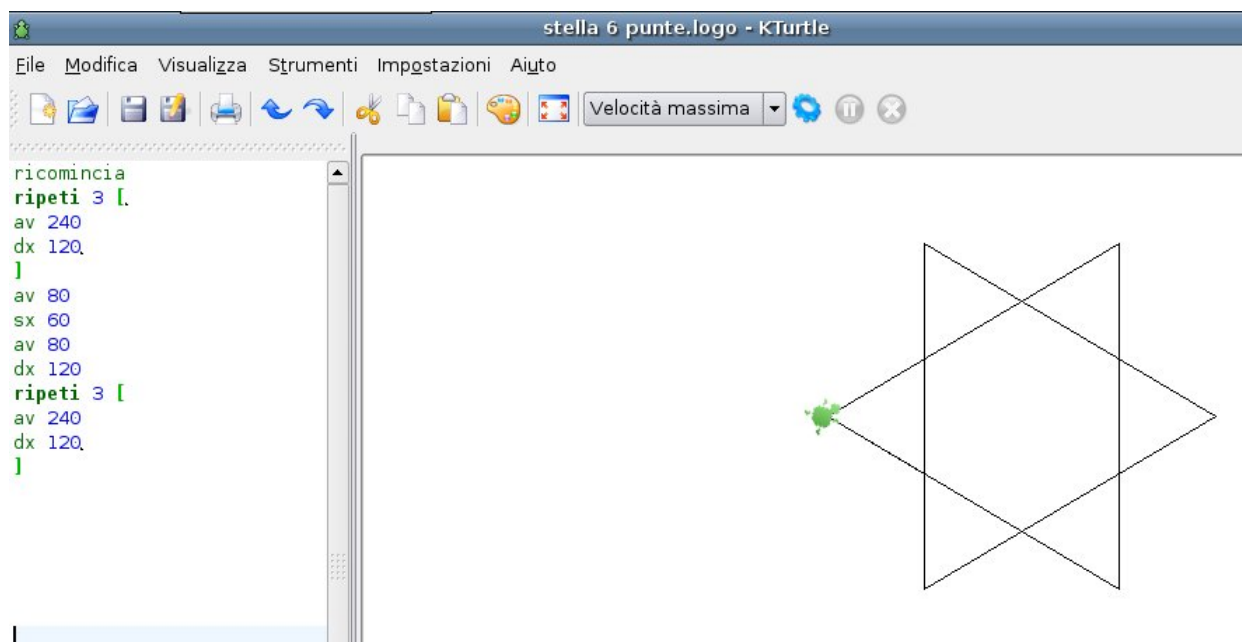
Un alunno, impostando lo sfondo blu e le stelle gialle, ha disegnato questo cielo stellato



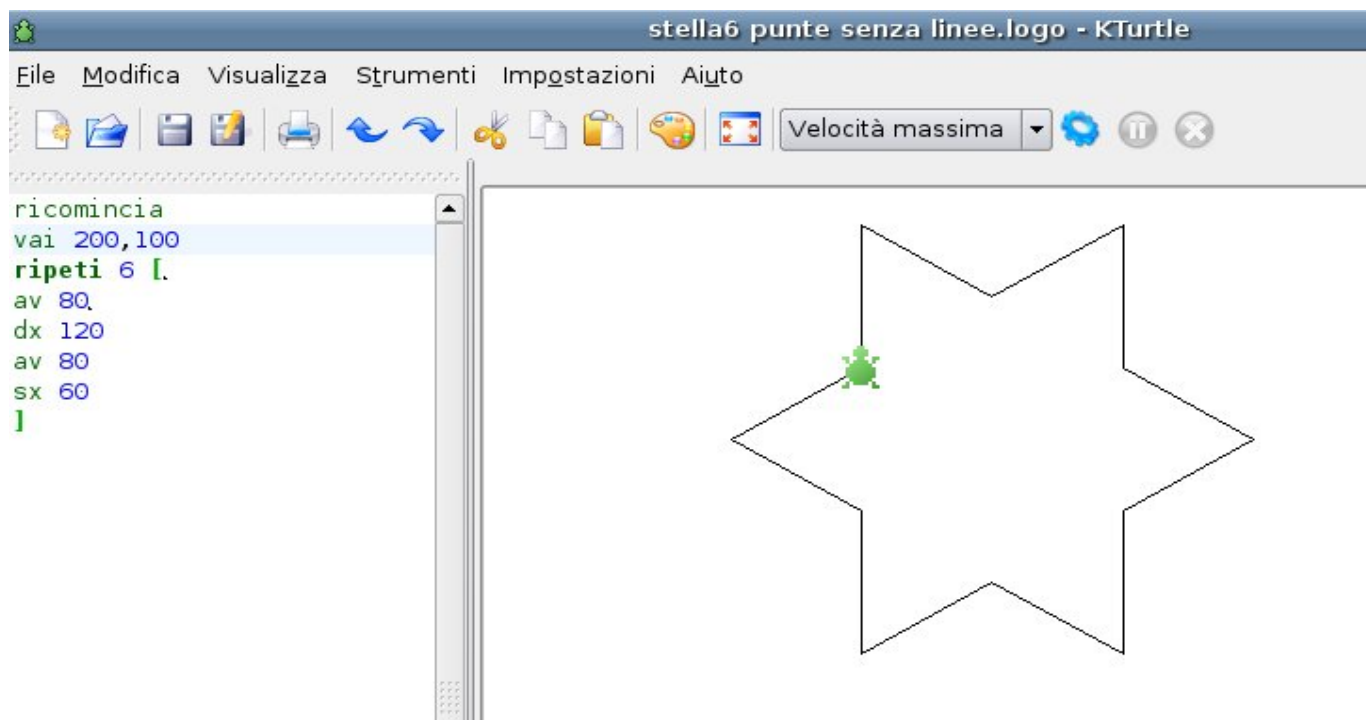
ALCUNE ESPERIENZE DIDATTICHE

Disegno di una stella a sei punte

Gli alunni osservando il disegno si accorgono che questa stella è formata da due triangoli equilateri e che ogni lato di un triangolo risulta diviso in tre parti uguali dai lati dell'altro.



Per disegnare una stella a sei punte, senza linee interne, gli alunni seguono il percorso della tartaruga, sapendo che la punta è un angolo con ampiezza di 60° .

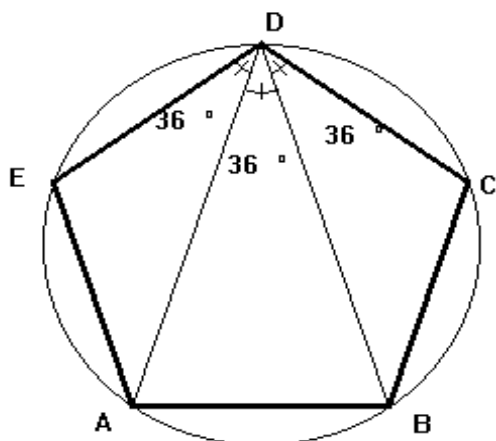


Entrambe le procedure possono essere scritte utilizzando le variabili:

Stella a 6 punte	Stella a sei punte senza linee interne
ricomincia l=240 ripeti 3 [av 1 dx 120] av 1/3 sx 60 av 1/3 dx 120 ripeti 3[av 1 dx 120]	ricomincia ripeti 6 [av 80 dx 120 av 80 sx 60]

Disegno di una sella a cinque punte

Più difficile risulta il disegno delle stelle a cinque punte perché non è facile stabilire l'ampiezza dell'angolo rappresentato da ogni singola punta.



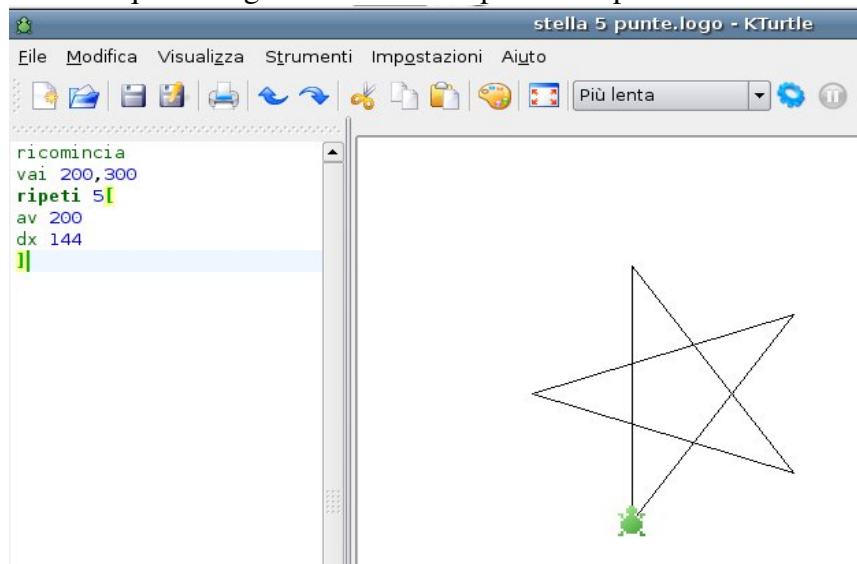
ANGOLO INTERNO DEL PENTAGONO EDC = 108°

**EDA = ADB = BDC
angoli uguali perché insistono su corde uguali)**

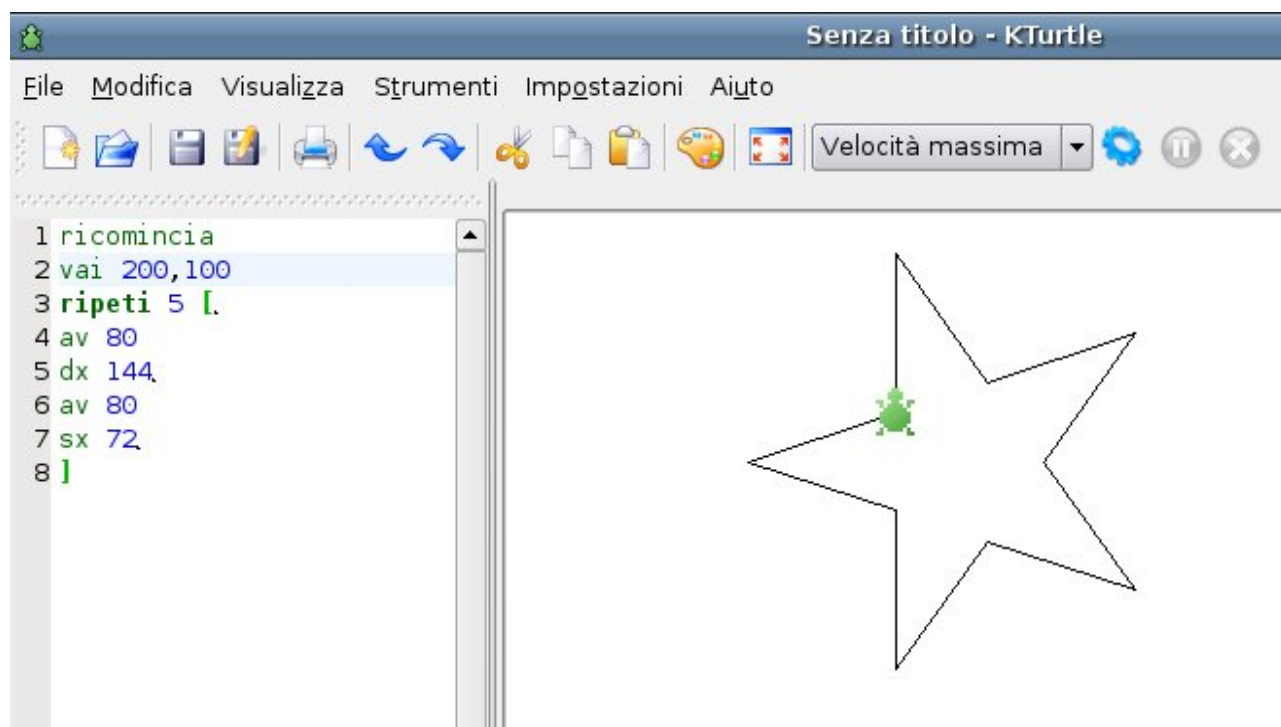
ANGOLO DELLA PUNTA DELLA STELLA

ADC = $108^\circ : 3 = 36^\circ$

Attraverso questo disegno, seguendo i vari passaggi, gli alunni comprendono che ogni punta ha un'ampiezza di 36° . Sono quindi in grado di trovare la procedura per realizzare il disegno:



e quello senza linee interne:



Anche in questo caso entrambe le procedure possono essere scritte utilizzando le variabili:

Stella a 5 punte	Stella a 5 punte senza linee interne
ricomincia	ricomincia
l = 200	a = 80
ripeti 5[ripeti 5 [
av l	av a
dx 144	dx 144
]	av a
	sx 72
]

Pavimentazioni

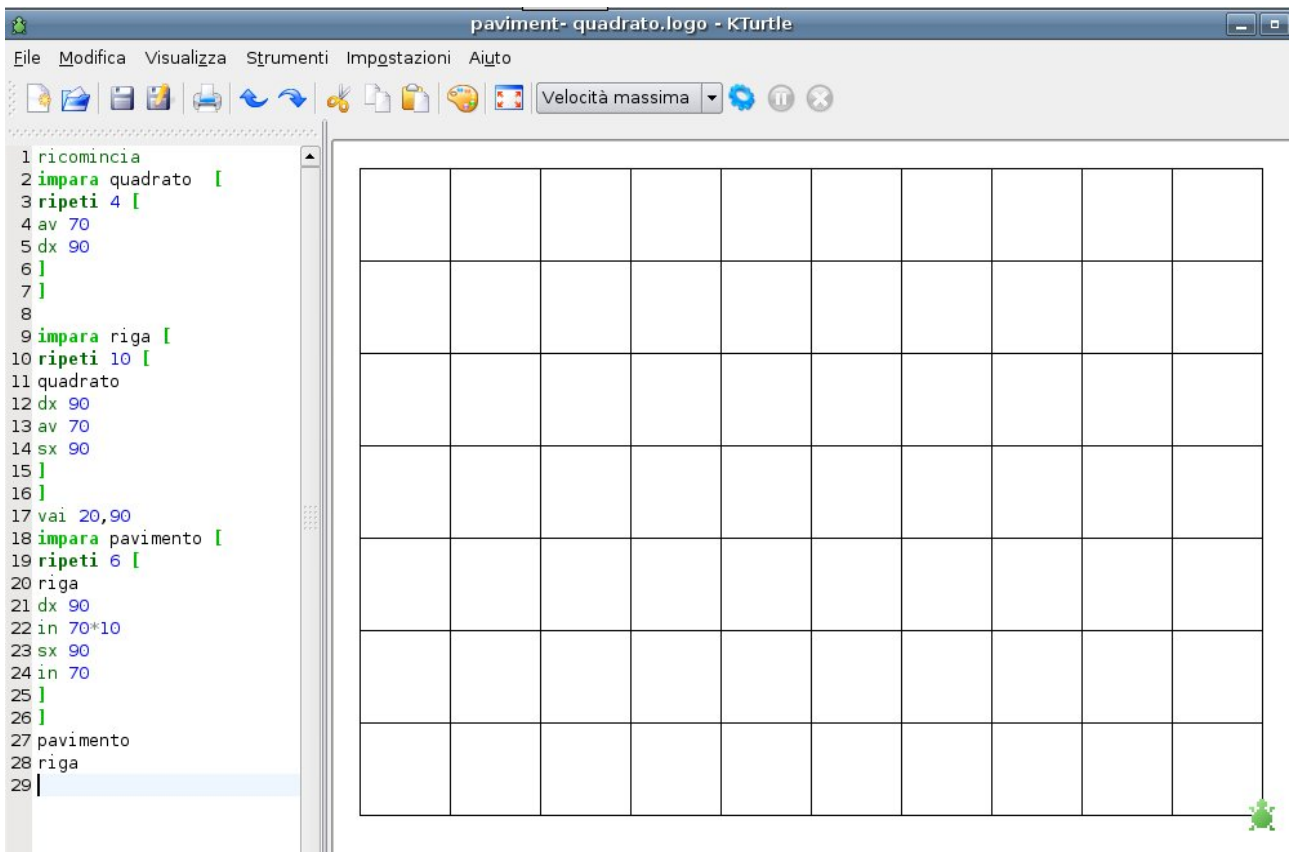
Un'attività che ha coinvolto in modo positivo gli alunni facendo sviluppare loro fantasia nel trovare soluzioni alternative, è stata quella della rappresentazioni di pavimentazioni.

Iniziando dalle più semplici sono arrivati a realizzare procedure sempre più complesse.

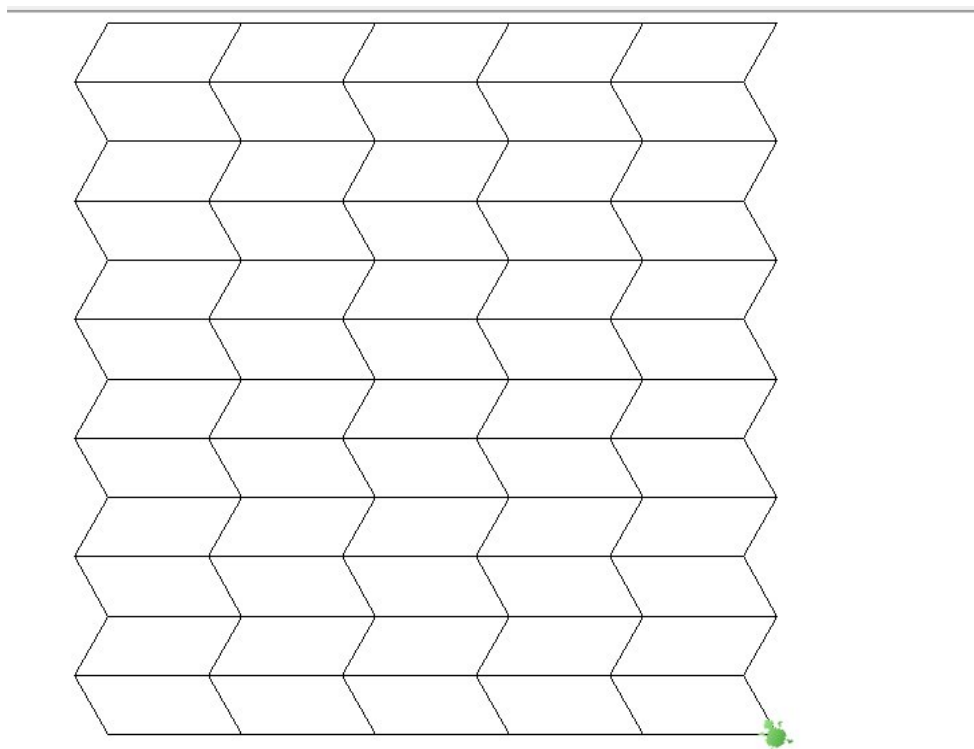
Nelle pavimentazioni non si sono introdotte variabili perché la variazione del lato delle singole figure fa variare la loro dimensione, mentre la dimensione dell'area di disegno rimane fissa.

Analizziamo alcune pavimentazioni realizzate dagli alunni; ricopriamo l'area di disegno con :

quadrati



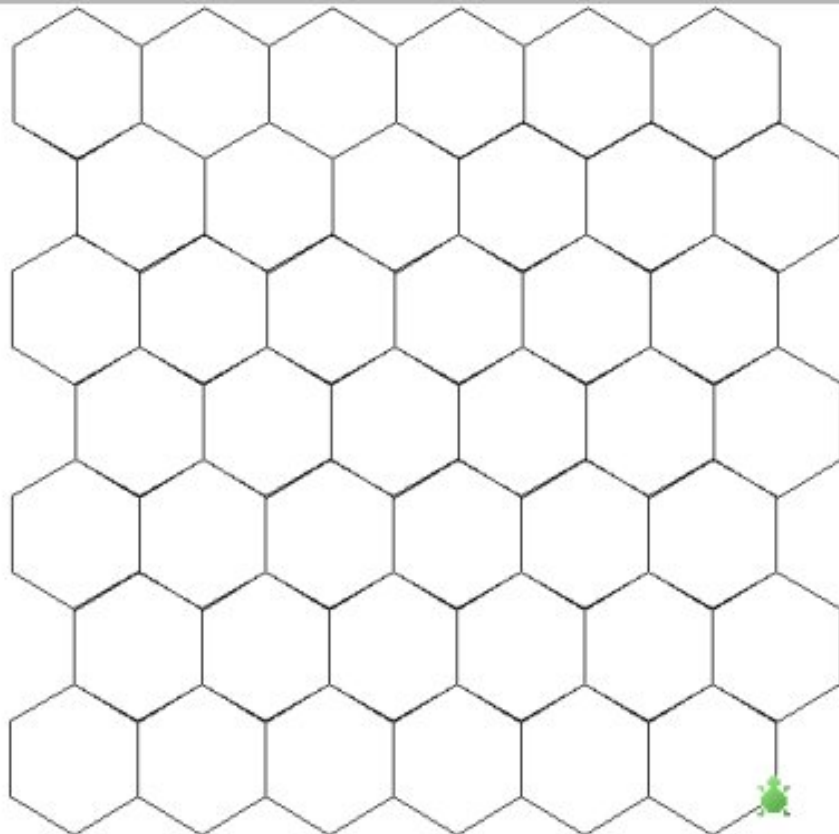
parallelogrammi disposti a lisca di pesce



parallelogrammi disposti a lisca di pesce : alla fine della prima colonna si passa alla seconda

ricomincia impari para1 [ripeti 2 [av 50 dx 60 av 100 dx 120]] vai 50,50 dx 30 impari riga1 [ripeti 5 [para1 dx 60 av 100 sx 60]]	impari para2 [ripeti 2 [av 50 dx 120 av 100 dx 60]] impari riga2 [ripeti 5 [para2 dx 120 av 100 sx 120]]	impari due righe [riga1 dx 60 in 500 dx 60 av 50 sx 180 riga2] impari pavimento [ripeti 5 [duerighe sx 60 av 500 dx 120 in 50] duerighe] pavimento
---	--	---

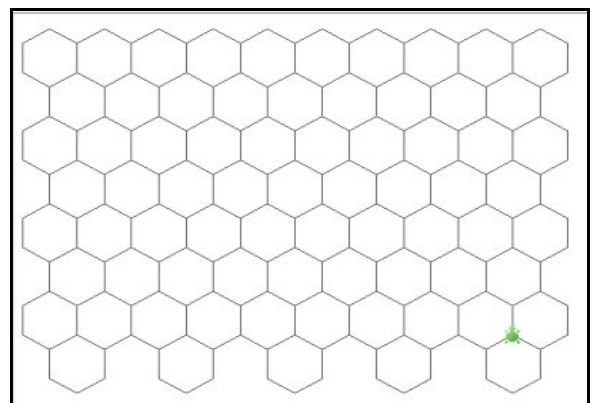
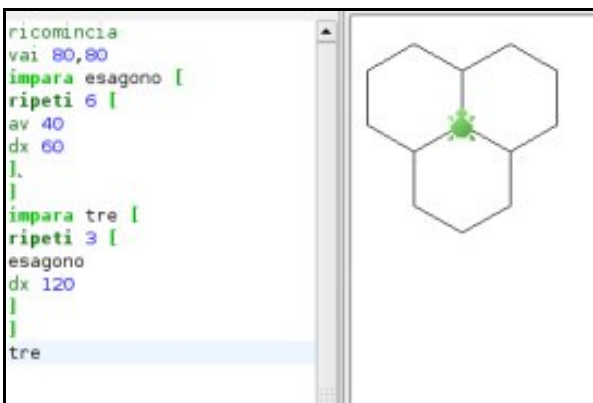
esagoni



<pre>ricomincia vai 50,70 impara riga [ripeti 6 [av 45 dx 60] dx 120 av 45 sx 60 av 45 sx 60]</pre>	<pre>ripeti 6 [riga] ripeti 3 [ps sx 90 av 71.5*6 dx 90 in 67 pg ripeti 6 [riga]</pre>	<pre>ps sx 90 av 84.5*6 dx 90 in 67 pg ripeti 6 [riga]</pre>
---	---	--

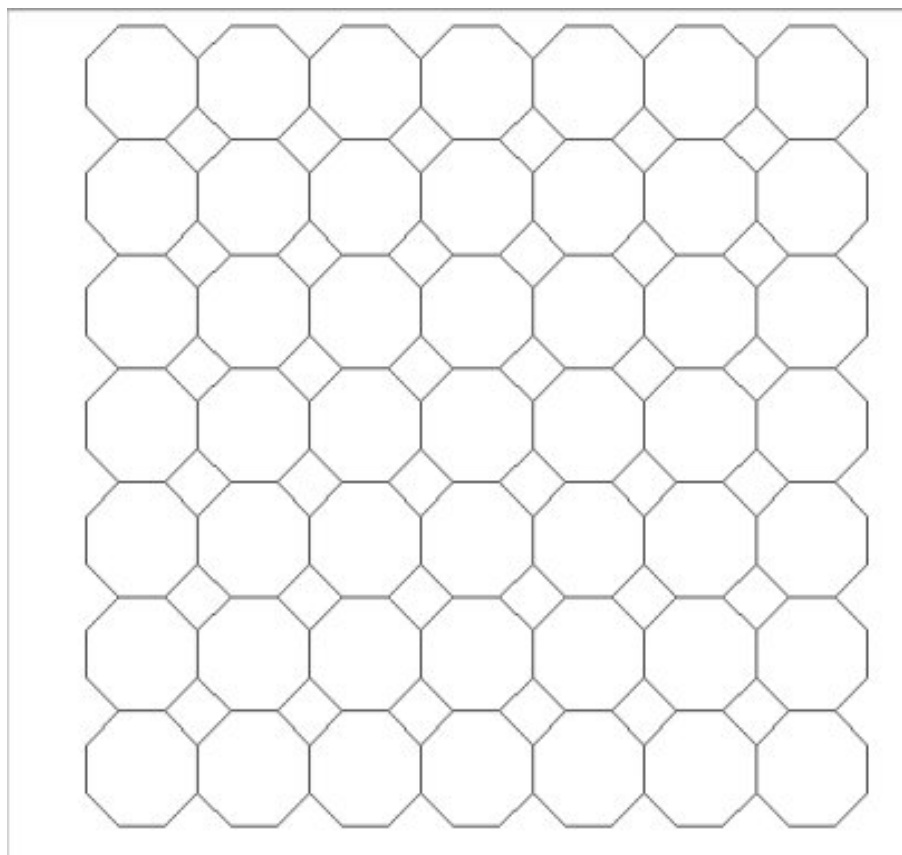
I comandi evidenziati sono stati trovati per approssimazioni successive, visto che gli alunni, di prima media, non conoscevano il teorema di Pitagora.

Un modo differente per una pavimentazione a esagoni è stato quello realizzato, dopo aver lavorato per un periodo sulle rotazioni, dalla rotazione di 120° di un esagono



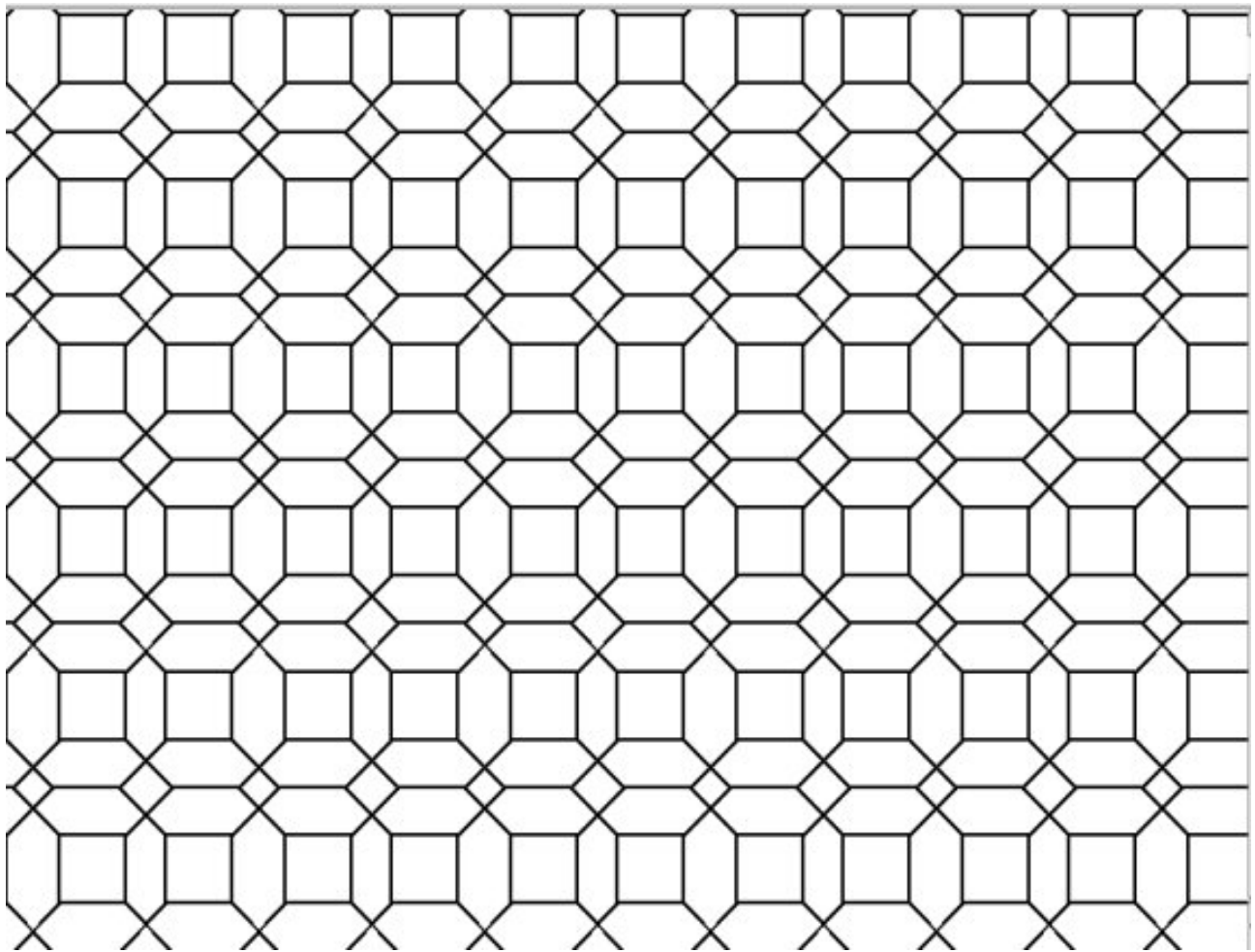
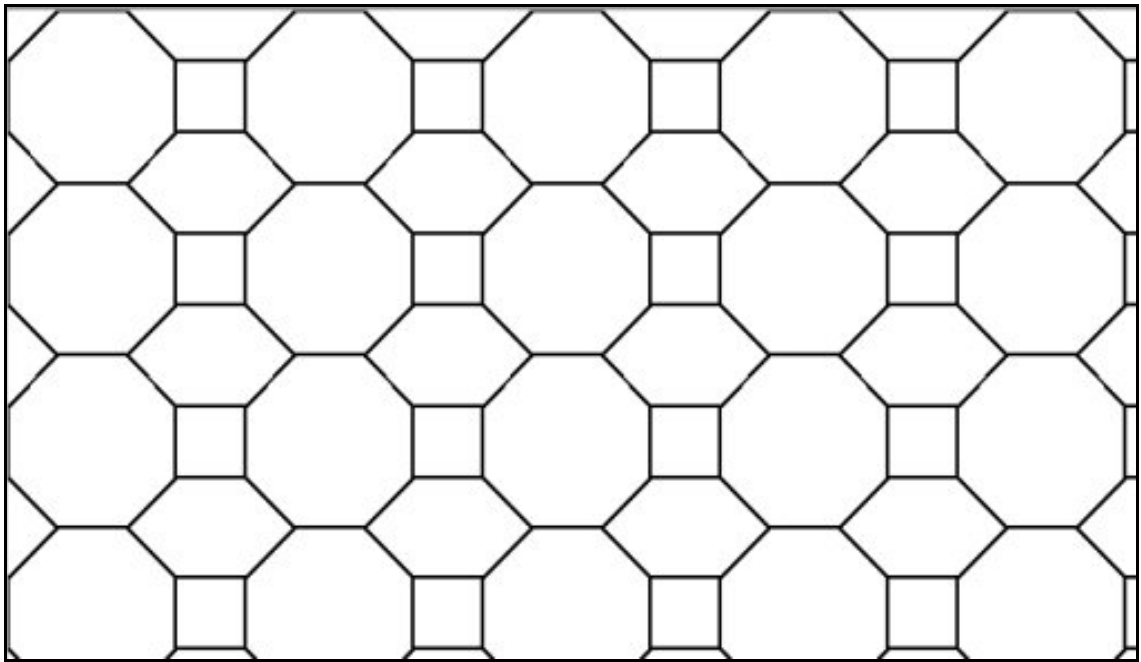
<pre>ricomincia vai 80,80 impara esagono [ripeti 6 [av 40 dx 60]] impara tre [ripeti 3 [esagono dx 120]] tre</pre>	<pre>]] impara riga [ripeti 4 [tre dx 120 av 40 sx 60 av 40 dx 60 av 40 sx 60</pre>	<pre>av 40 sx 60] tre] impara pavimento [ripeti 3 [riga dx 60 in 40 ripeti 8[dx 60 in 40</pre>	<pre>sx 60 in 40] sx 60 in 40 sx 60 in 40] riga] pavimento</pre>
--	--	---	---

ottagoni



ricomincia vai 50,60 impara ottagono [ripeti 8 [av 30 dx 45] dx 135 av 30 sx 45 av 30 sx 45 av 30 sx 45]]	impara riga [ripeti 7 [ottagono] ps ripeti 7 [sx 135 av 30 dx 45 av 30 dx 45 av 30 dx 45]]	dx 135 av 30 dx 90 av 30 dx 135 in 30 pg] ripeti 7 [riga]]
---	--	---

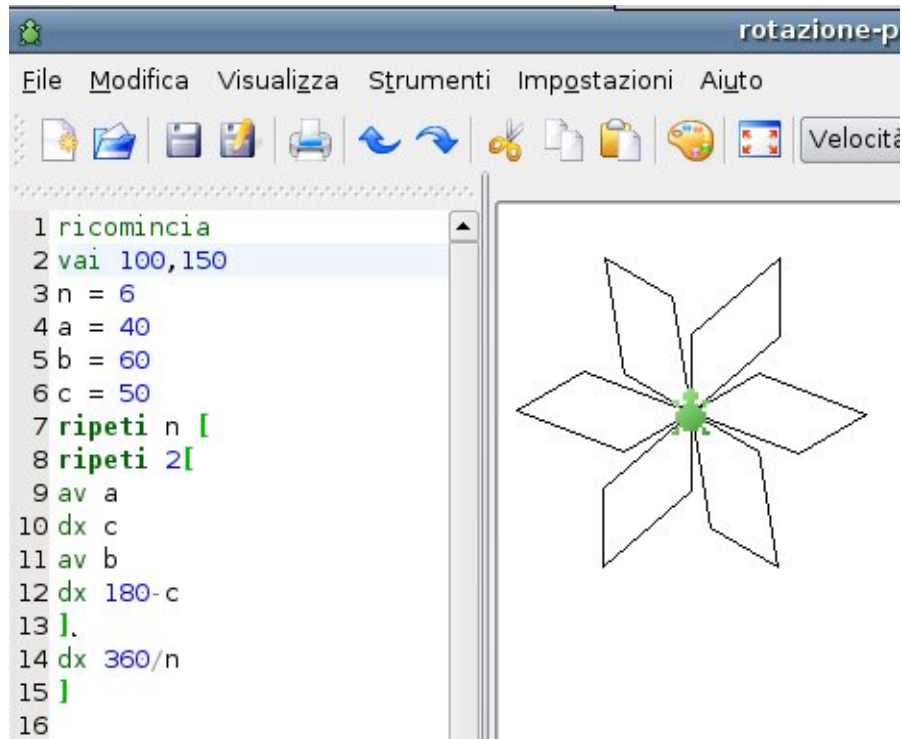
Altri tipi di pavimentazione realizzati dai ragazzi:



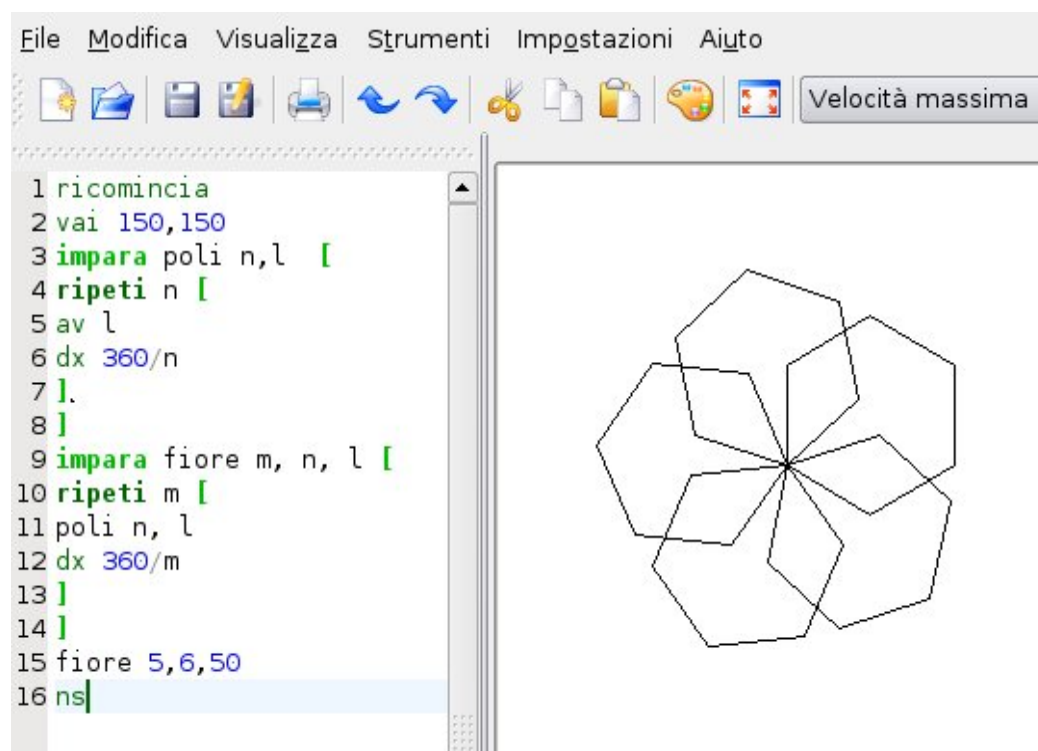
Rotazioni

Le figure si possono far ruotare rispetto ad un centro di rotazione. E' utile sul piano didattico riflettere su quale sia l'angolo di rotazione da assegnare perché la figura, dopo una rotazione di 360° , ritorni nella posizione iniziale. Gli alunni lo capiscono, come al solito, dall'osservazione dei disegni fatti da loro su carta, ma spesso tendono ad apprenderlo per tentativi, direttamente lavorando sul computer. L'angolo di rotazione sarà dato da $360/n$, dove n è il numero delle volte che viene rappresentata la figura.

Ecco un esempio di rotazione di un parallelogramma:



Estendendo la procedura ad un poligono regolare in cui le variabili sono l , misura del lato del poligono, n numero lati poligono e m numero di rotazioni:



Con *kturtle* si può anche scrivere sull'area di disegno

Se si vogliono **scrivere dei numeri** è sufficiente scrivere sull'area dei comandi **scrivi** seguito da un **numero**. Se l'immagine della tartaruga si sovrappone al testo, si può nasconderla scrivendo **ns** (nascondi) per poi farla ricomparire quando serve scrivendo **ms** (mostra)

Oltre a scrivere un numero il programma permette di scrivere anche *il risultato di una qualsiasi operazione* o di *espressioni numeriche*. Esempio se si scrive **scrivi 4*5** nell'area dei comandi, in quella di disegno comparirà **20**.

Se si vuole **scrivere un testo** si usa ancora il comando **scrivi**, ma il **testo va scritto tra virgolette**.

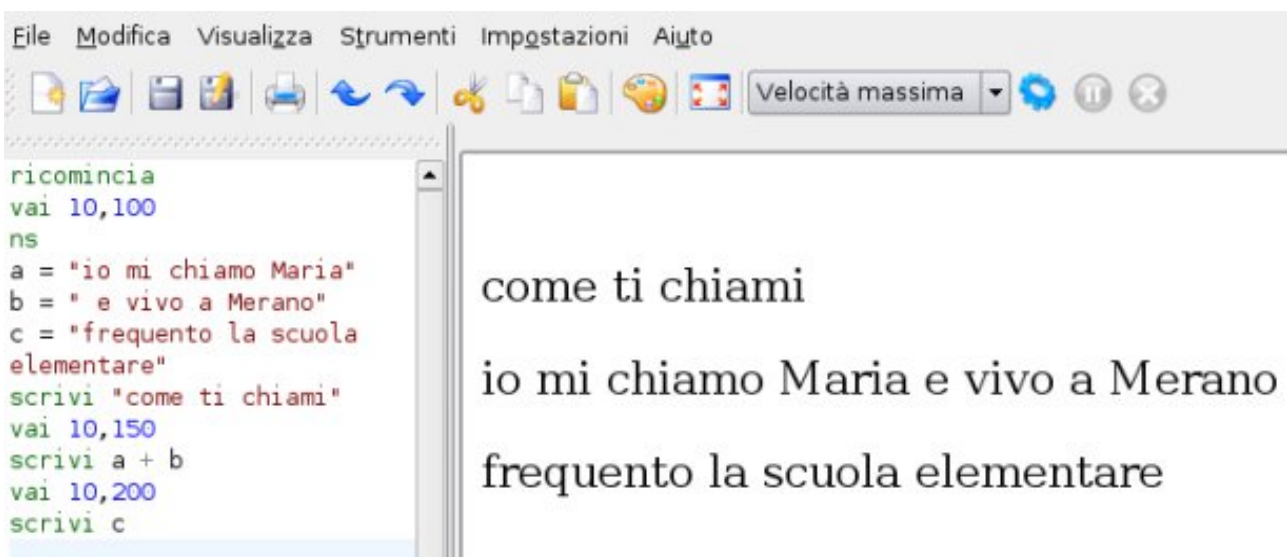
Si deve tenere conto della lunghezza del testo rispetto alla dimensione dello schermo ed usare il comando **vai x,y** per posizionare in modo opportuno la frase.

Si possono usare delle "stringhe", simili alle variabili alle quali si attribuisce come valore una frase.

Come si vede dall'esempio riportato se voglio che sull'area di disegno mi compaiano le scritte che ho chiamato a, b, c è sufficiente scrivere **scrivi a** se voglio che venga scritta la frase "io mi chiamo Maria", **scrivi a + b** se voglio che vengano scritte in sequenza le frasi "io mi chiamo Maria" e "e vivo a Merano".

Se, nell'esempio sotto riportato, si vuole che tra "io mi chiamo Maria" e "e vivo a Merano" ci sia uno spazio, si deve lasciare uno spazio tra le virgolette e l'inizio o la fine della frase. Esempio:

a = "io mi chiamo Maria" b = "e vivo a Merano", oppure a = "io mi chiamo Maria" b = " e vivo a Merano"



```
ricomincia
vai 10,100
ns
a = "io mi chiamo Maria"
b = " e vivo a Merano"
c = "frequento la scuola
elementare"
scrivi "come ti chiami"
vai 10,150
scrivi a + b
vai 10,200
scrivi c
```

come ti chiami

io mi chiamo Maria e vivo a Merano

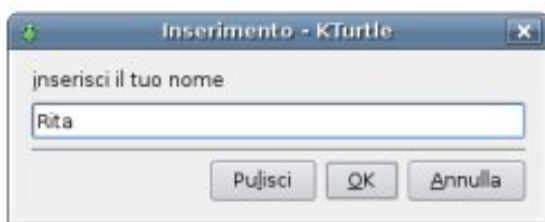
frequento la scuola elementare

Finestrainserimento

Il comando `finestrainserimento` chiede di introdurre una domanda o un dato che viene poi utilizzato dal programma in un testo o, se è un numero, in un'operazione.

Per esempio in questa procedura:

```
Ricomincia
vai 100, 100
a = " ciao "
nome = finestrainserimento "inserisci il tuo nome"
scrivi a + nome + ", come va? Come si chiama il tuo cane?"
vai 100, 200
c = finestrainserimento "come si chiama il tuo cane"
b = "il mio cane che si chiama "
scrivi b + c + " e' buono"
```

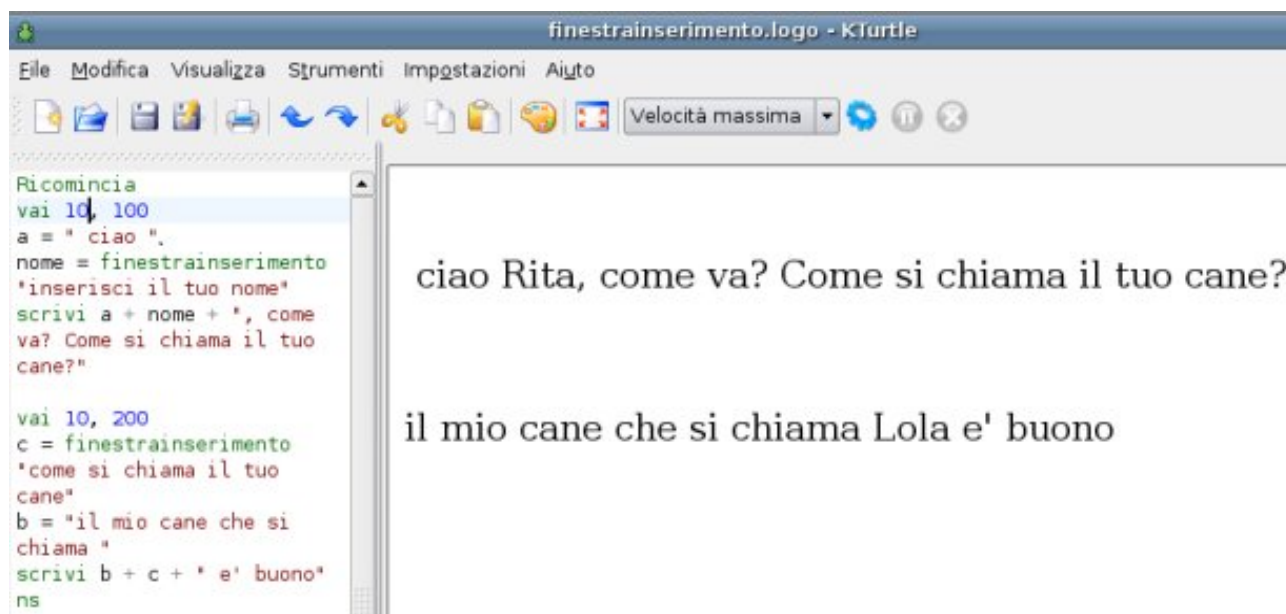


il comando `nome=finestrainserimento "inserisci il tuo nome"` è una domanda che compare quando si esegue la procedura, mentre la risposta "Rita" viene scritta nell'area di disegno.

Nello stesso modo la domanda `c = finestrainserimento "come si chiama il tuo cane"` non verrà scritta nell'area

di disegno apparirà la risposta "Lola".

Come prima, le stringhe sono collegate tra loro , o con un testo tra virgolette, con un +.



Problemi impostati con l'utilizzo “finestrainserimento”

Esempio di risoluzione di problema per il calcolo del perimetro e dell'area di un rettangolo, con l'uso di finestrainserimento per dare i valori della base e dell'altezza:

ricomincia

ns

vai 10, 50

scrivi "Calcola il perimetro e l'area di un rettangolo."

b = finestrainserimento "scrivi la base "

h = finestrainserimento "scrivil'altezza "

vai 10,100

$p = (b+h)*2$

scrivi "il perimetro e' " + p

vai 10,150

$k = b * h$

scrivi "l'area e' " + k

vai 300,250

ripeti 2

[

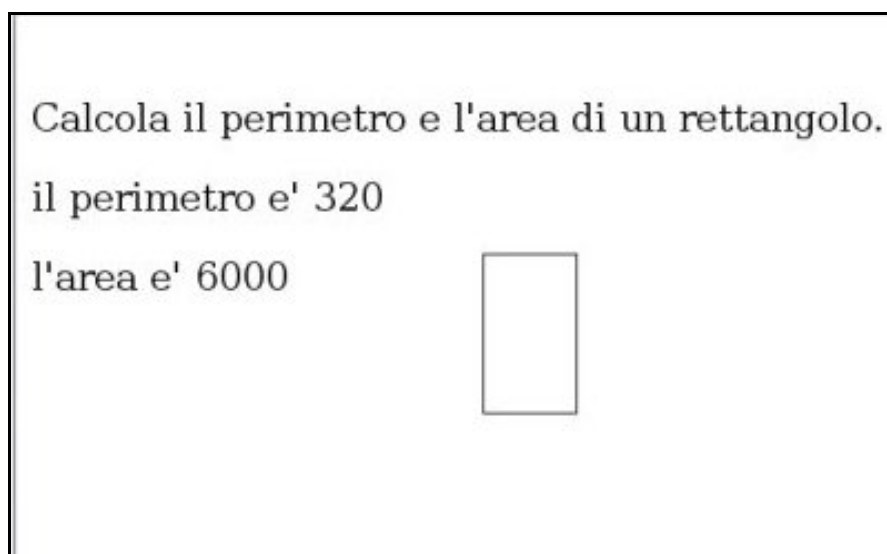
av h

dx 90

av b

dx 90

]



I valori della base e dell'altezza vengono inseriti, ma non compaiono nell'area di disegno e servono al programma solo per calcolare il perimetro ed l'area del rettangolo.

In questa procedura, scritta da un alunno, è rappresentato anche il rettangolo in scala.

Procedure con controllo del risultato

Oltre a scrivere procedure che permettono di ottenere risultati di operazioni e di problemi di ogni tipo inserendo nella procedura stessa l'algoritmo risolutivo, è anche possibile creare procedure che siano in grado di richiedere il risultato e controllare se sia corretto.

Per esempio, dati due addendi, viene richiesta la loro somma. Se il risultato è corretto si potrà dare un messaggio positivo, in caso contrario negativo.

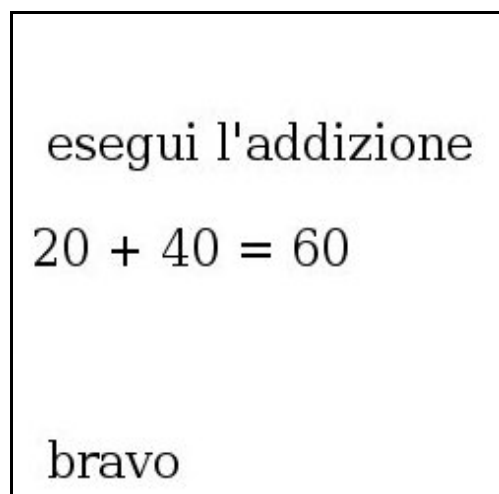
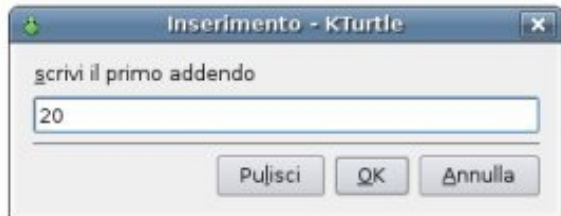
Procedura per addizione con controllo del risultato:

ricomincia

```
ns
vai 10,50
scrivi " esegui l'addizione "
a=finestrainserimento "scrivi il primo addendo"

b=finestrainserimento "scrivi il secondo
addendo"
risposta=finestrainserimento "esegui la somma"
somma=a+b
vai 10,100
scrivi a + " + " +b + " = " + risposta
vai 10,200
se somma == risposta [
scrivi " bravo "
] altrimenti [
scrivi " riprova "
]
```

esegui l'addizione



Se la *somma* = a+b è uguale alla *risposta* (valore che si ottiene da finestrainserimento= esegui la somma) i comandi da usare per stabilire se la risposta è corretta o errata sono:

```
se somma == risposta[
scrivi " bravo "
] altrimenti [
scrivi " riprova "
]
```

