

Informatica

e Tecnologie della Comunicazione Digitale

Docente:

Miguel Ceriani (ceriani@di.uniroma1.it)

Lezioni:

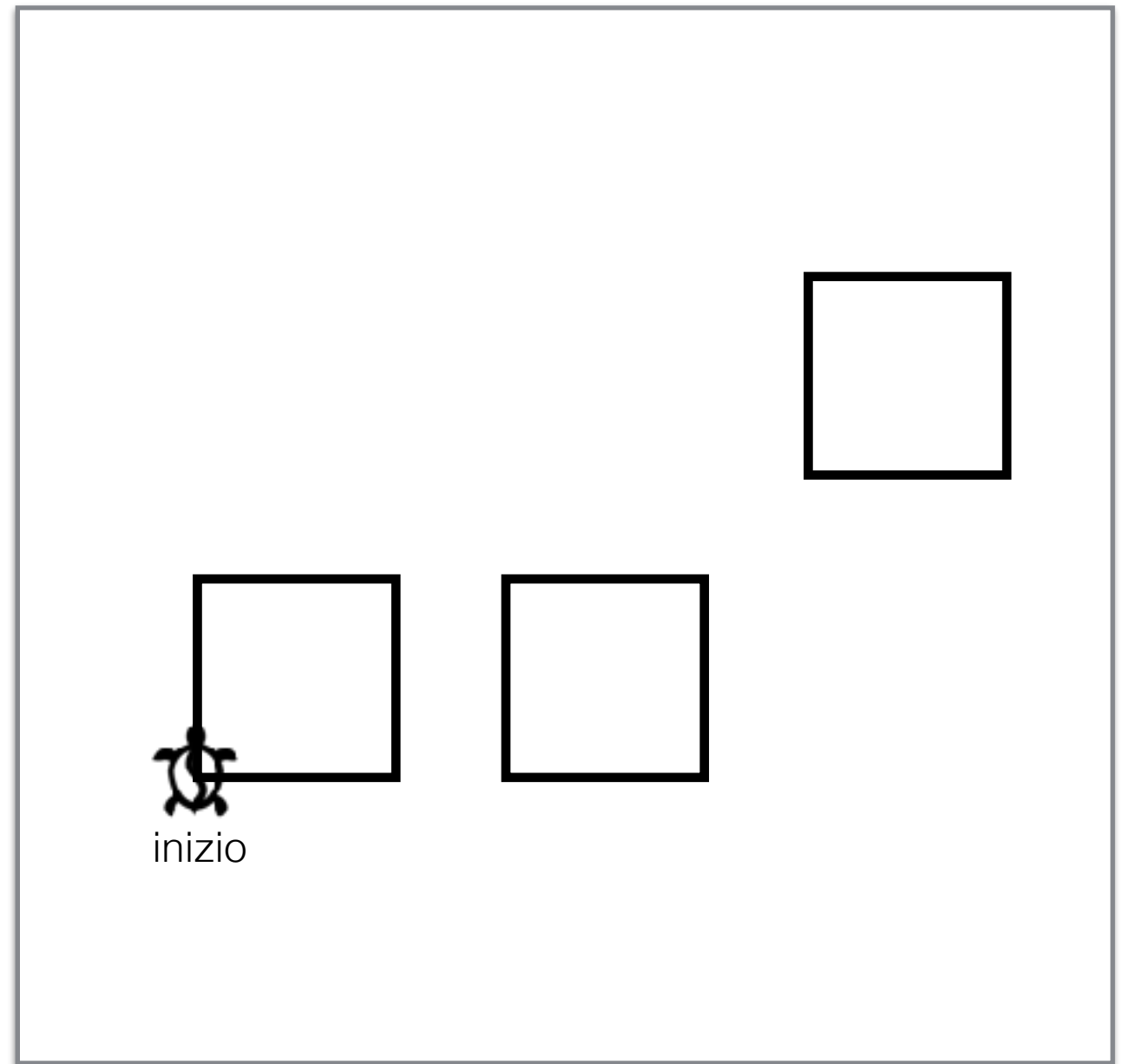
Mercoledì/Giovedì/Venerdì 9-11

Ricevimento (su appuntamento):

Mercoledì 14-16 a viale Regina Elena 295, palazzina F, 1° piano

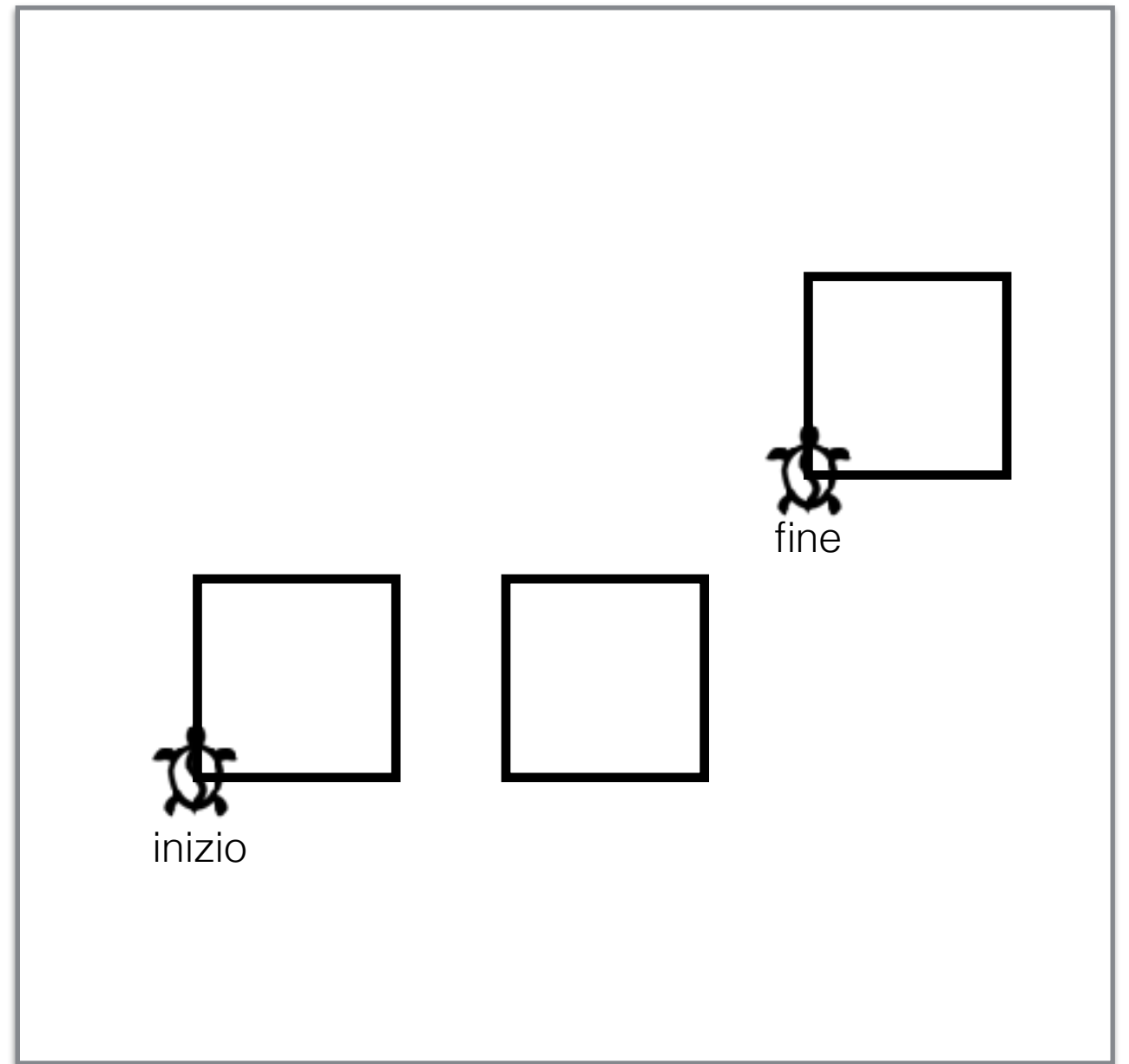
Lezione 12: Linguaggi di Programmazione: Funzioni/Procedure

Ancora Quadrati



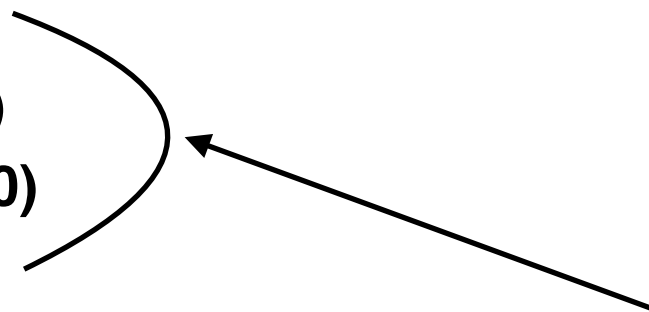
Ancora Quadrati

```
RIPETI(2) {  
  PENNA_GIU  
  RIPETI(4) {  
    AVANTI(20)  
    DESTRA(90)  
  }  
  PENNA_SU  
  DESTRA(90)  
  AVANTI(30)  
}  
SINISTRA(90)  
AVANTI(30)  
PENNA_GIU  
RIPETI(4) {  
  AVANTI(20)  
  DESTRA(90)  
}
```



Ancora Quadrati

```
RIPETI(2) {  
  PENNA_GIU  
  RIPETI(4) {  
    AVANTI(20)  
    DESTRA(90)  
  }  
  PENNA_SU  
  DESTRA(90)  
  AVANTI(30)  
}  
SINISTRA(90)  
AVANTI(30)  
PENNA_GIU  
RIPETI(4) {  
  AVANTI(20)  
  DESTRA(90)  
}
```

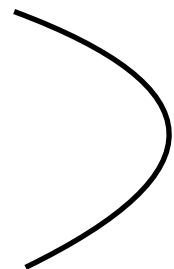
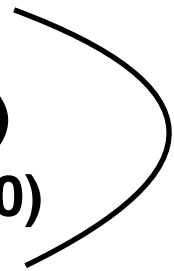


qui grazie al RIPETI
riesco ad evitare di
riscrivere istruzioni

Ancora Quadrati

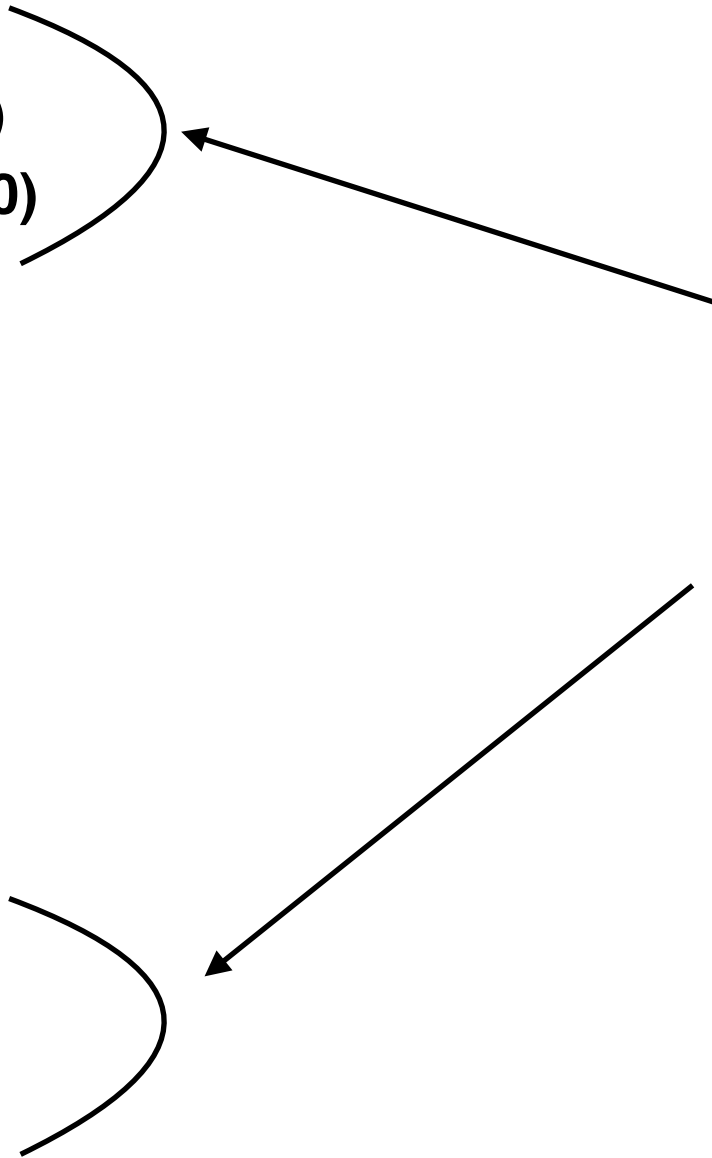
```
RIPETI(2) {  
  PENNA_GIU  
  RIPETI(4) {  
    AVANTI(20)  
    DESTRA(90)  
  }  
  PENNA_SU  
  DESTRA(90)  
  AVANTI(30)  
}  
SINISTRA(90)  
AVANTI(30)  
PENNA_GIU  
RIPETI(4) {  
  AVANTI(20)  
  DESTRA(90)  
}
```

però sono costretto a
ripetere due volte le
istruzioni che generano
un quadrato!



Ancora Quadrati

```
RIPETI(2) {  
  PENNA_GIU  
  RIPETI(4) {  
    AVANTI(20)  
    DESTRA(90)  
  }  
  PENNA_SU  
  DESTRA(90)  
  AVANTI(30)  
}  
SINISTRA(90)  
AVANTI(30)  
PENNA_GIU  
RIPETI(4) {  
  AVANTI(20)  
  DESTRA(90)  
}
```



posso dare un nome a questa sequenza (ad es., *quadrato*) e poi usare solo il nome tutte le volte che mi serve?

Procedure/Funzioni

- quando ho una sequenza di istruzioni a cui voglio dare un nome per riusarla più volte, parlo di *procedura* o *funzione* (nel seguito funzione)
- il *nome* della funzione è la stringa che riuserò come etichetta per usarla
- il *corpo* della funzione è la sequenza di istruzioni
- la *definizione* della funzione è l'istruzione che “crea” quella funzione, definendone nome e corpo
- una *chiamata* a una funzione è ogni volta che uso la funzione, attraverso il suo nome

Funzioni: Sintassi

- **DEF** *nome_funzione* {
 istruzione1
 istruzione2
 ...
}

dove *nome_funzione* è una stringa di testo e *istruzione1*, *istruzione2*, ... sono istruzioni del linguaggio (ad es.

AVANTI(...), DESTRA(...), RIPETI(...) {...}, ...)

- *nome_funzione*

dove *nome_funzione* è il nome di una funzione precedentemente definita con DEF

Funzioni: Semantica

- **DEF** *nome_funzione* {
 istruzione1
 istruzione2
 ...
}

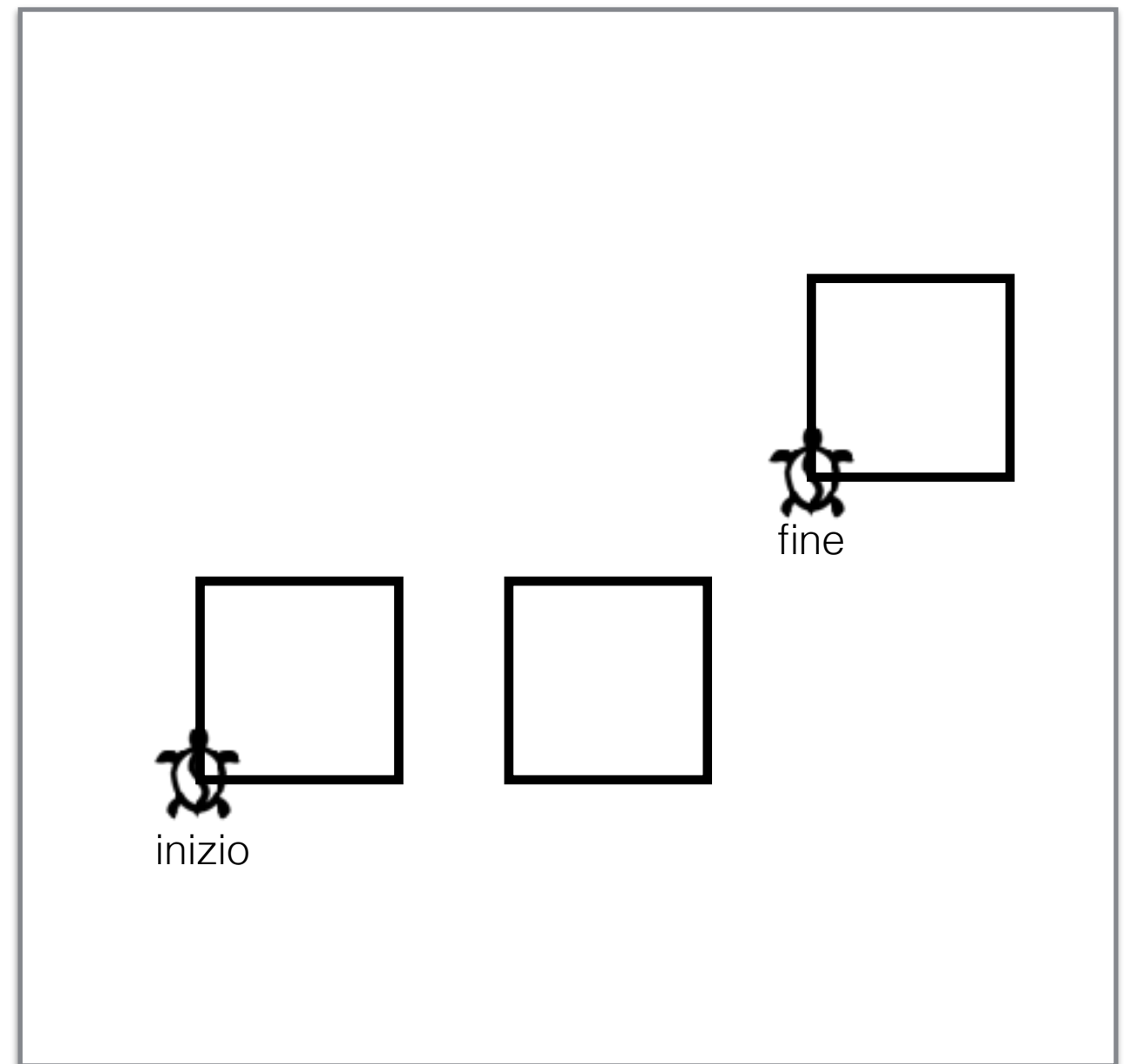
definisce una nuova funzione, il cui nome è *nome_funzione* e il corpo è la sequenza *istruzione1*, *istruzione2*, ...

- *nome_funzione*
esegue una chiamata alla (precedentemente definita) funzione di nome *nome_funzione*; la chiamata consiste l'esecuzione del corpo della funzione

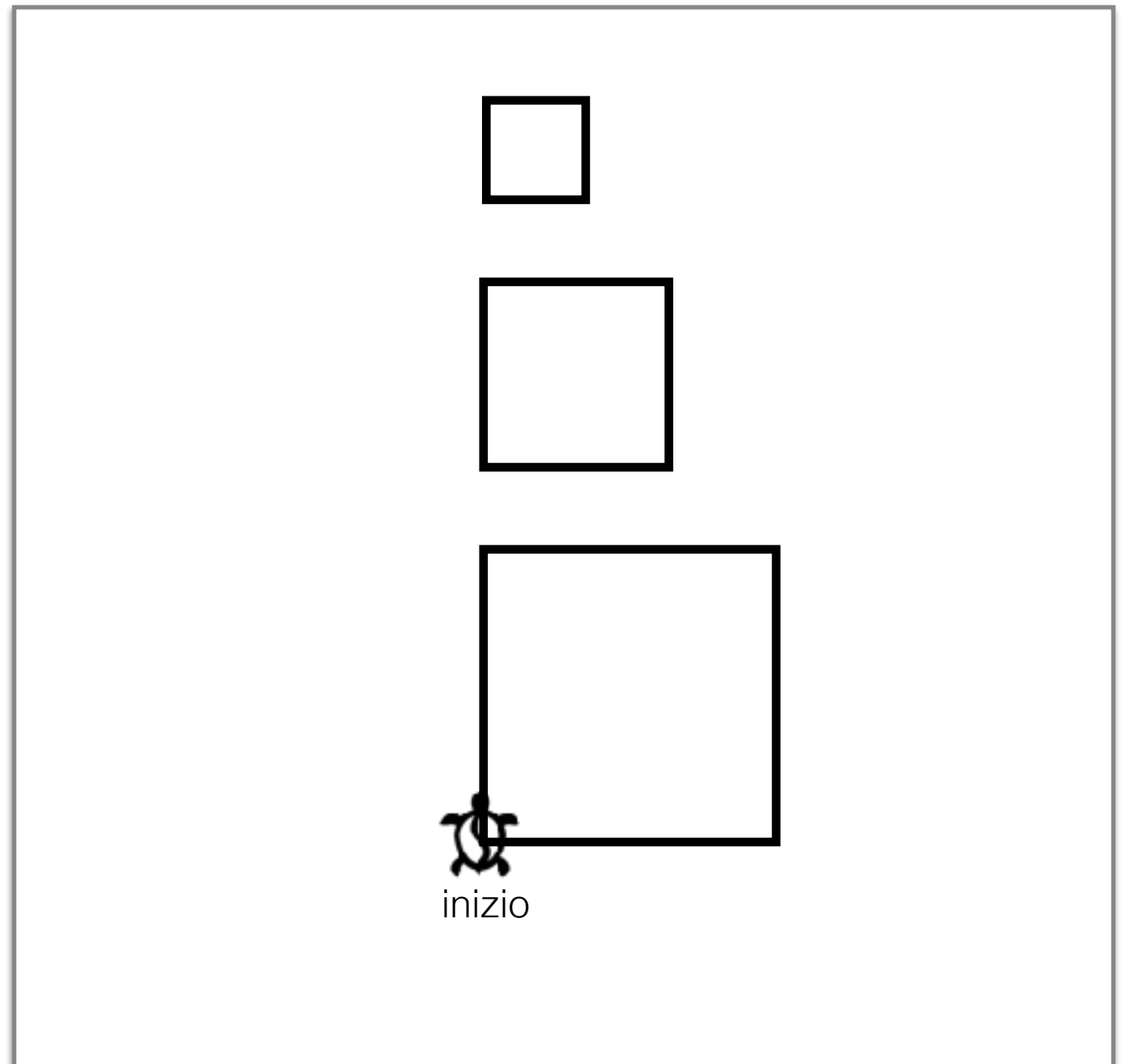
Programma con Definizione e Uso di una Funzione

```
DEF quadrato {  
  PENNA_GIU  
  RIPETI(4) {  
    AVANTI(20)  
    DESTRA(90)  
  }  
  PENNA_SU  
}
```

```
RIPETI(2) {  
  quadrato  
  DESTRA(90)  
  AVANTI(30)  
}  
SINISTRA(90)  
AVANTI(30)  
quadrato
```

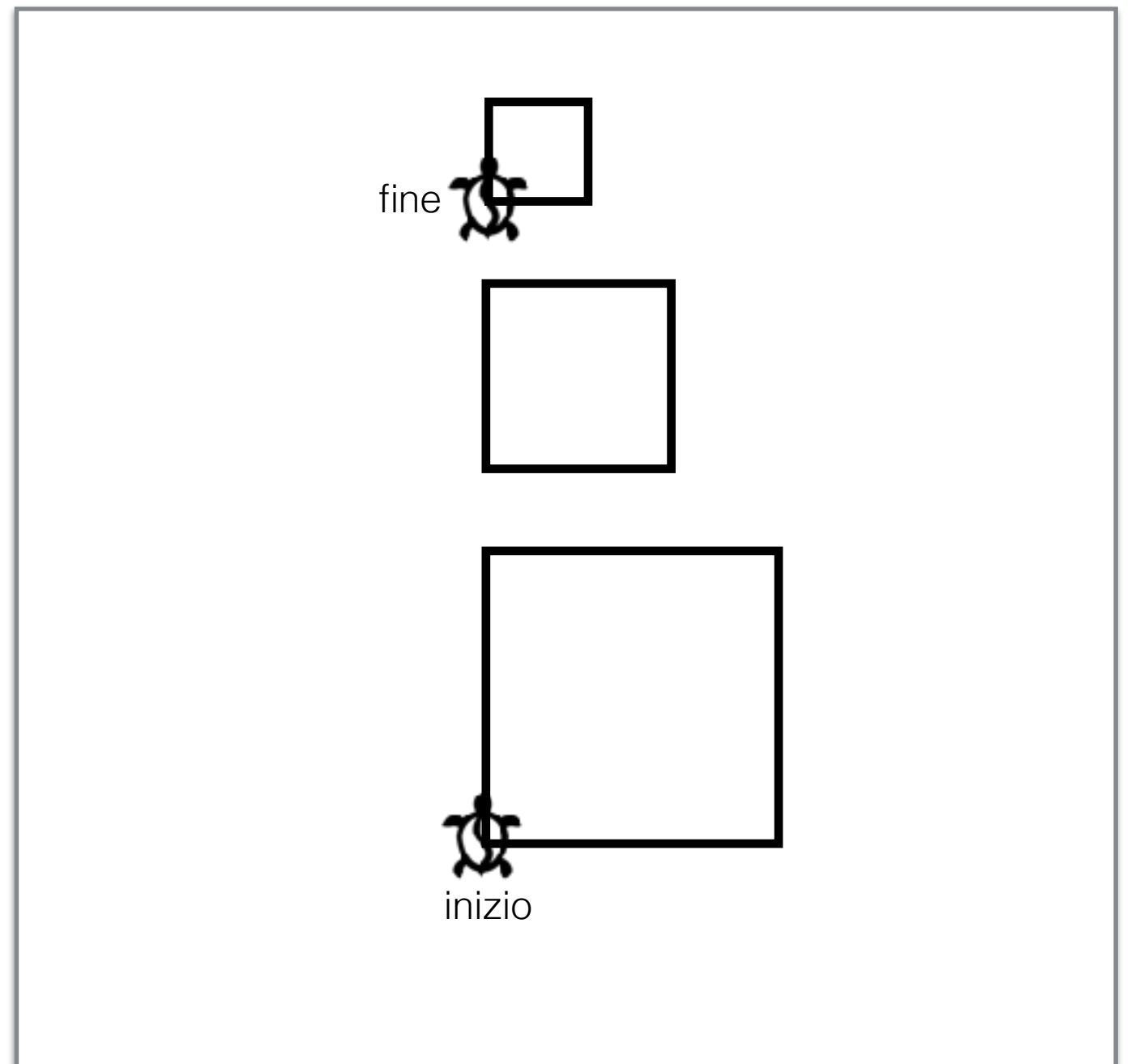


Quadrati Diversi



Quadrati Diversi

```
RIPETI(4) {  
  AVANTI(30)  
  DESTRA(90)  
}  
PENNA_SU  
AVANTI(40)  
PENNA_GIU  
RIPETI(4) {  
  AVANTI(20)  
  DESTRA(90)  
}  
PENNA_SU  
AVANTI(30)  
PENNA_GIU  
RIPETI(4) {  
  AVANTI(10)  
  DESTRA(90)  
}  
}
```



Quadrati Diversi

```
RIPETI(4) {  
  AVANTI(30)  
  DESTRA(90)  
}
```

```
PENNA_SU  
AVANTI(40)  
PENNA_GIU
```

```
RIPETI(4) {  
  AVANTI(20)  
  DESTRA(90)  
}
```

```
PENNA_SU  
AVANTI(30)  
PENNA_GIU
```

```
RIPETI(4) {  
  AVANTI(10)  
  DESTRA(90)  
}
```

le istruzioni per creare i quadrati sono molto simili (cambia solo il valore utilizzato per AVANTI)!!!

Quadrati Diversi

```
RIPETI(4) {  
  AVANTI(30)  
  DESTRA(90)  
}
```

```
PENNA_SU  
AVANTI(40)  
PENNA_GIU
```

```
RIPETI(4) {  
  AVANTI(20)  
  DESTRA(90)  
}
```

```
PENNA_SU  
AVANTI(30)  
PENNA_GIU
```

```
RIPETI(4) {  
  AVANTI(10)  
  DESTRA(90)  
}
```

posso definire un'unica
funzione che possa
essere utilizzata per
quadrati di diversa
grandezza?

Funzioni con Parametri

- mi serve definire funzioni che siano un po' più “flessibili”: il comportamento possa essere caratterizzato a seconda delle necessità (per es., per la funzione quadrato che si possa decidere la lunghezza del lato)
- nella definizione della funzione, dichiaro uno o più *parametri*, ovvero valori che la funzione accetterà
- nel corpo della funzione potrò utilizzare i nomi dei parametri, che verranno poi sostituiti dai valori passati alla funzione
- nella chiamata della funzione, per ogni parametro dichiarato, ho un'espressione, il cui valore verrà passato alla funzione

Funzioni con Parametri: Sintassi

- **DEF** *nome_funzione*(*nome_param1*, *nome_param2*, ...) {
 istruzione1
 istruzione2
 ...
}

dove *nome_funzione*, *nome_param1*, *nome_param2*, ... sono stringhe di testo e *istruzione1*, *istruzione2*, ... sono istruzioni del linguaggio (ad es. **AVANTI(...)**, **DESTRA(...)**, **RIPETI(...)** {...}, ...)

- *nome_funzione*(*espr1*, *espr2*, ...)

dove *nome_funzione* è il nome di una funzione precedentemente definita con DEF e *espr1*, *espr2*, ... sono espressioni

Funzioni con Parametri: Semantica

- **DEF** *nome_funzione*(*nome_param1*, *nome_param2*, ...) {
 istruzione1
 istruzione2
 ...
}

definisce una nuova funzione, il cui nome è *nome_funzione* e il corpo è la sequenza *istruzione1*, *istruzione2*, ...

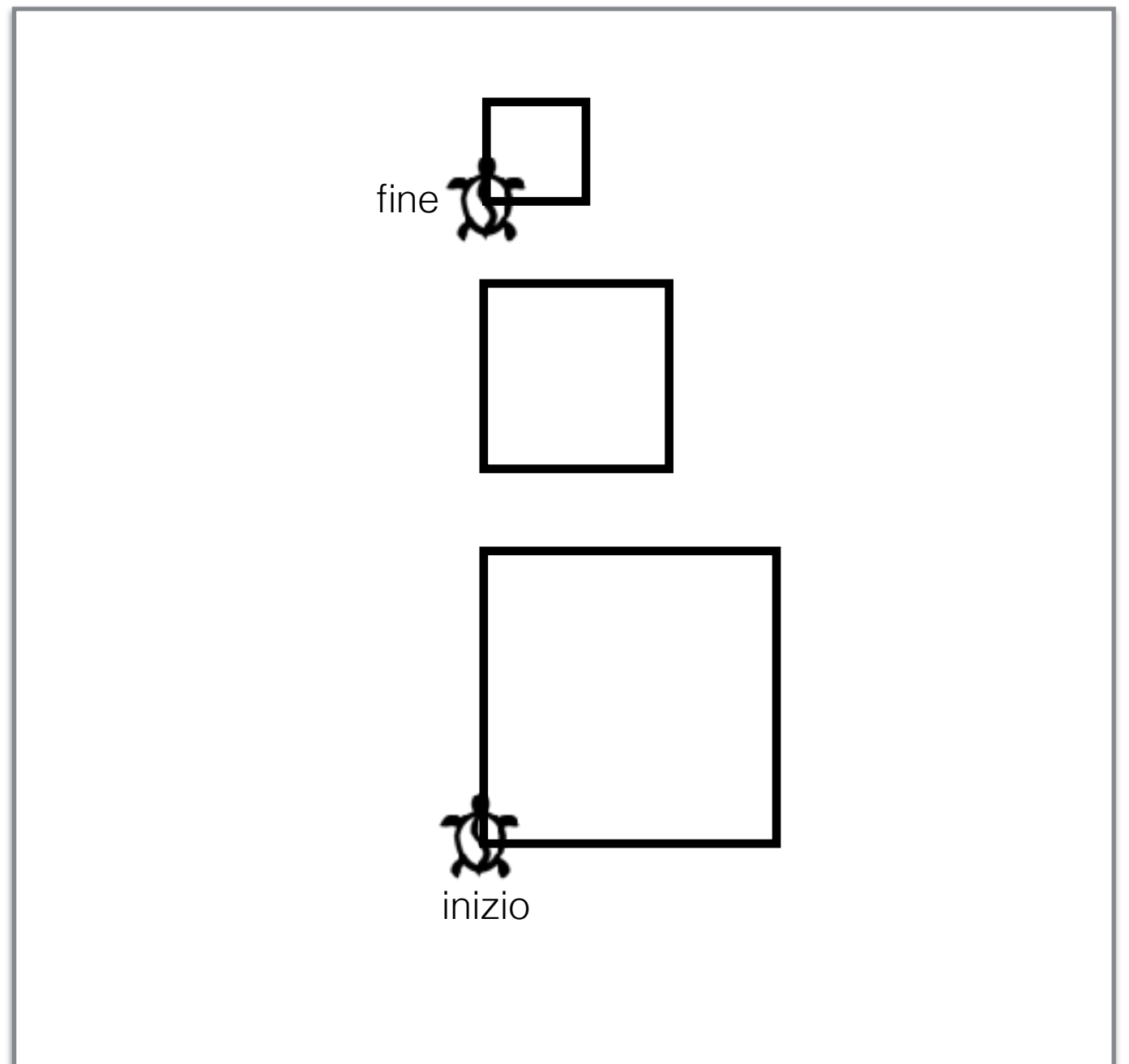
- *nome_funzione*(*espr1*, *espr2*, ...)

esegue una chiamata alla (precedentemente definita) funzione di nome *nome_funzione*, passando per ogni parametro il valore ottenuto dalla corrispondente espressione; la chiamata consiste l'esecuzione del corpo della funzione, in cui ogni occorrenza di ciascun parametro è sostituita dal valore passato.

Quadrati Diversi

```
DEF quadrato(lato) {  
  PENNA_GIU  
  RIPETI(4) {  
    AVANTI(lato)  
    DESTRA(90)  
  }  
  PENNA_SU  
}
```

```
quadrato(30)  
AVANTI(40)  
quadrato(20)  
AVANTI(30)  
quadrato(10)
```



Unificazione Sintassi

per uniformità usiamo le
parentesi anche per le
funzioni senza parametri

```
DEF quadrato() {  
    PENNA_GIU  
    RIPETI(4) {  
        AVANTI(20)  
        DESTRA(90)  
    }  
    PENNA_SU  
}
```

```
RIPETI(2) {  
    quadrato()  
    DESTRA(90)  
    AVANTI(30)  
}  
SINISTRA(90)  
AVANTI(30)  
quadrato()
```