

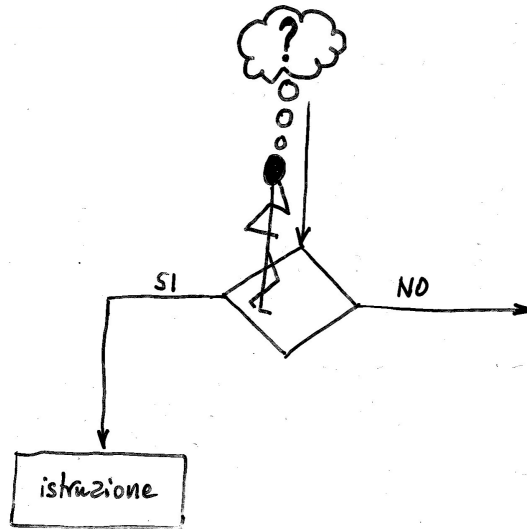
Strutture di controllo

Corso di Programmazione I A, 2021-22
Felice Cardone

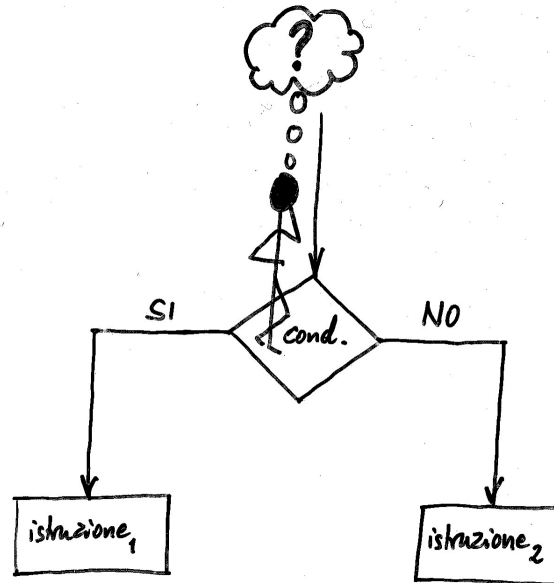
Istruzione condizionale

Istruzione condizionale

Condizionale a una via:



Condizionale a due vie:



Istruzione condizionale

Formato generale

Condizionale a una via:

```
if (<espressione booleana>)  
    <istruzione>
```

Condizionale a due vie:

```
if (<espressione booleana>)  
    <istruzione1>  
else  
    <istruzione2>
```

Esecuzione di un'istruzione condizionale a una via

1. si valuta l'espressione booleana;
2. (a) se la valutazione termina e si ottiene il valore true:
 si esegue la istruzione
(b) se la valutazione termina e si ottiene il valore false:
 si passa alla (eventuale) prossima istruzione del programma

Istruzione condizionale

Formato generale

Condizionale a una via:

```
if (<espressione booleana>)  
    <istruzione>
```

Condizionale a due vie:

```
if (<espressione booleana>)  
    <istruzione1>  
else  
    <istruzione2>
```

Esecuzione di un'istruzione condizionale a due vie

1. si valuta l'espressione booleana;
2. (a) se la valutazione termina e si ottiene il valore true:
 si esegue la istruzione₁

 (b) se la valutazione termina e si ottiene il valore false:
 si esegue la istruzione₂

Esempi

Determinare quale valore viene stampato su output dopo l'esecuzione della seguente istruzione:

```
if (2 < 3)  
    System.out.println("uno");  
else  
    System.out.println("due");
```

Esempi

Determinare quale valore viene stampato su output dopo l'esecuzione della seguente istruzione:

```
if (2 < 3)  
    System.out.println("uno");  
else  
    System.out.println("due");
```

uno

Esempi

Determinare quale valore viene stampato su output dopo l'esecuzione della seguente istruzione:

```
if (3 < 2)  
    System.out.println("uno");  
else  
    System.out.println("due");
```


Esempi

Determinare quale valore viene stampato su output dopo l'esecuzione della seguente istruzione:

```
if (3 < 2)  
    System.out.println("uno");  
else  
    System.out.println("due");
```

due

Esempi

Determinare quale valore viene stampato su output dopo l'esecuzione della seguente istruzione:

```
if (2 < 3)  
    System.out.println("uno");  
System.out.println("due");
```

Esempi

Determinare quale valore viene stampato su output dopo l'esecuzione della seguente istruzione:

```
if (2 < 3)  
    System.out.println("uno");  
System.out.println("due");
```

uno
due

Esempi

Determinare quale valore viene stampato su output dopo l'esecuzione della seguente istruzione:

```
if (3 < 2)  
    System.out.println("uno");  
System.out.println("due");
```

Esempi

Determinare quale valore viene stampato su output dopo l'esecuzione della seguente istruzione:

```
if (3 < 2)  
    System.out.println("uno");  
System.out.println("due");
```

due

Istruzioni vs sequenze di istruzioni: blocchi

Una sequenza di istruzioni non è una istruzione!

Il frammento di codice

```
if (3 < 2)
    System.out.println("uno");
System.out.println("due");
```

contiene due istruzioni: un condizionale a una via, seguito da una istruzione di scrittura su output.

Per creare una istruzione a partire da una sequenza di istruzioni

$\text{istruzione}_1; \text{istruzione}_2; \dots ; \text{istruzione}_n$

si forma il blocco $\{\text{istruzione}_1; \text{istruzione}_2; \dots ; \text{istruzione}_n\}$.

Allora l'esecuzione di

```
if (3 < 2) {
    System.out.println("uno");
    System.out.println("due");
}
```

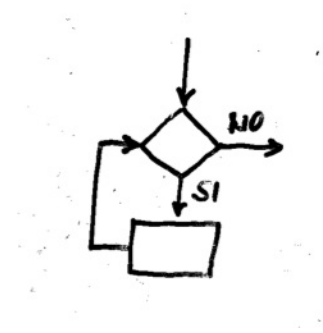
non ha effetto.

Iterazione: istruzione while

Istruzione while

Formato generale

while (<espressione booleana>)
 <istruzione>



Esecuzione di un'istruzione while

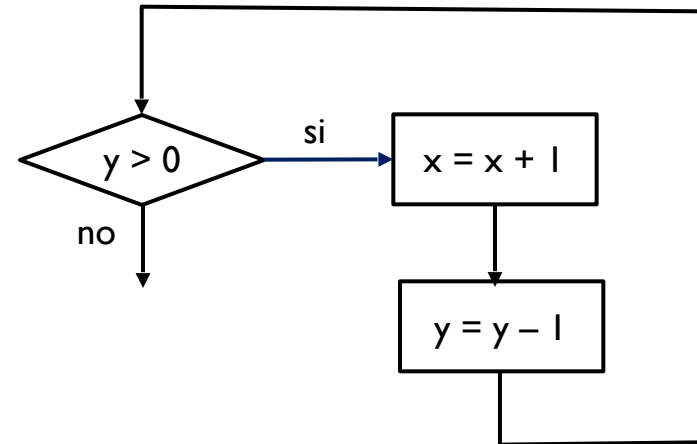
1. si valuta l'espressione booleana;
2. (a) se la valutazione termina e si ottiene il valore true:
 si esegue la istruzione e si torna al passo 1
- (b) se la valutazione termina e si ottiene il valore false:
 si prosegue con la (eventuale) prossima istruzione del programma

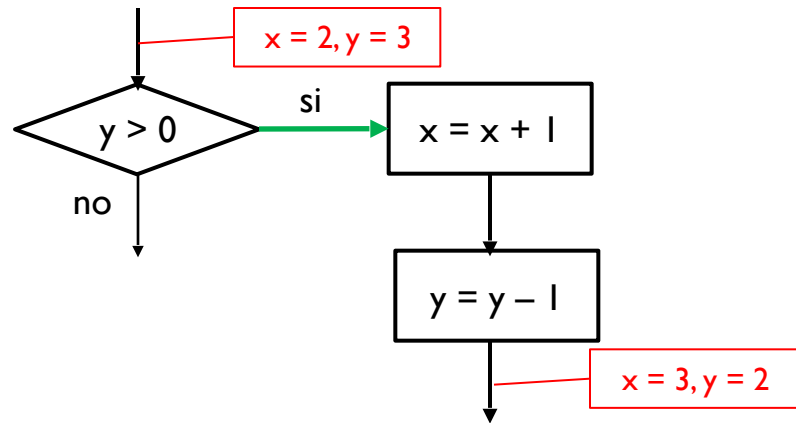
Istruzione while: esempio

```
x = 2; y = 3;  
while (y > 0) {  
    x = x + 1;  
    y = y - 1;  
}
```

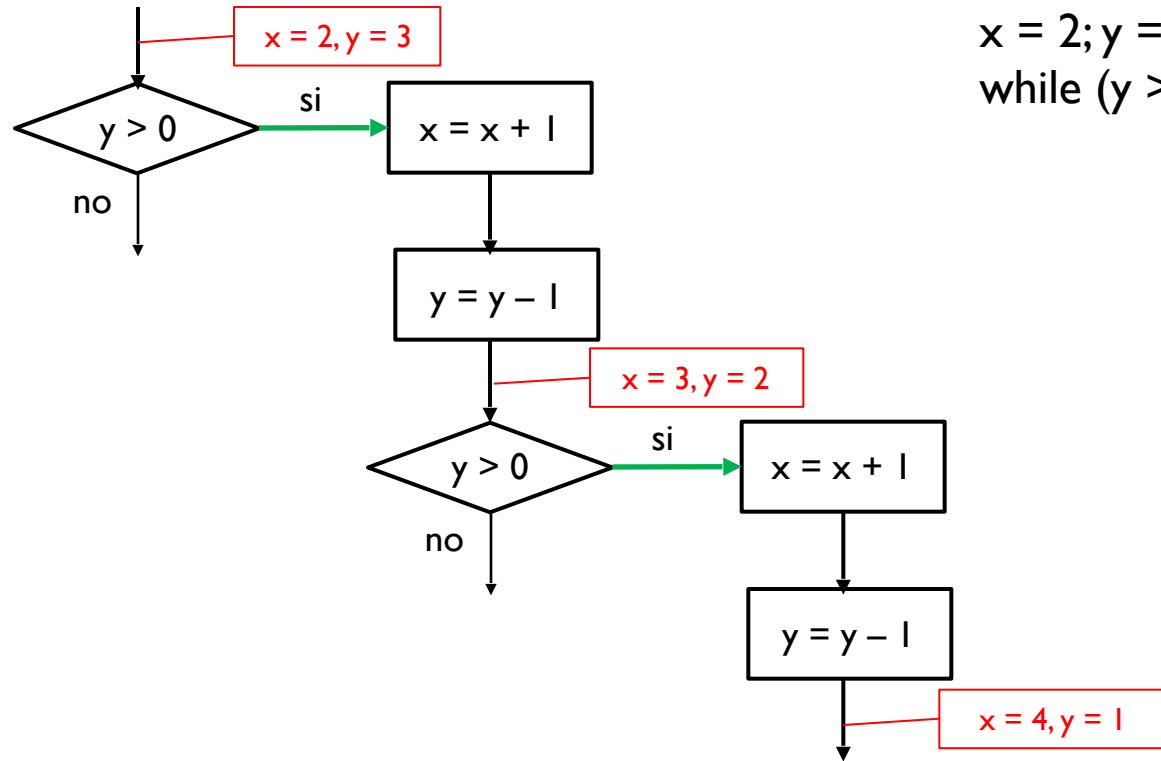
Istruzione while: esempio

```
x = 2; y = 3;  
while (y > 0) {  
    x = x + 1;  
    y = y - 1;  
}
```

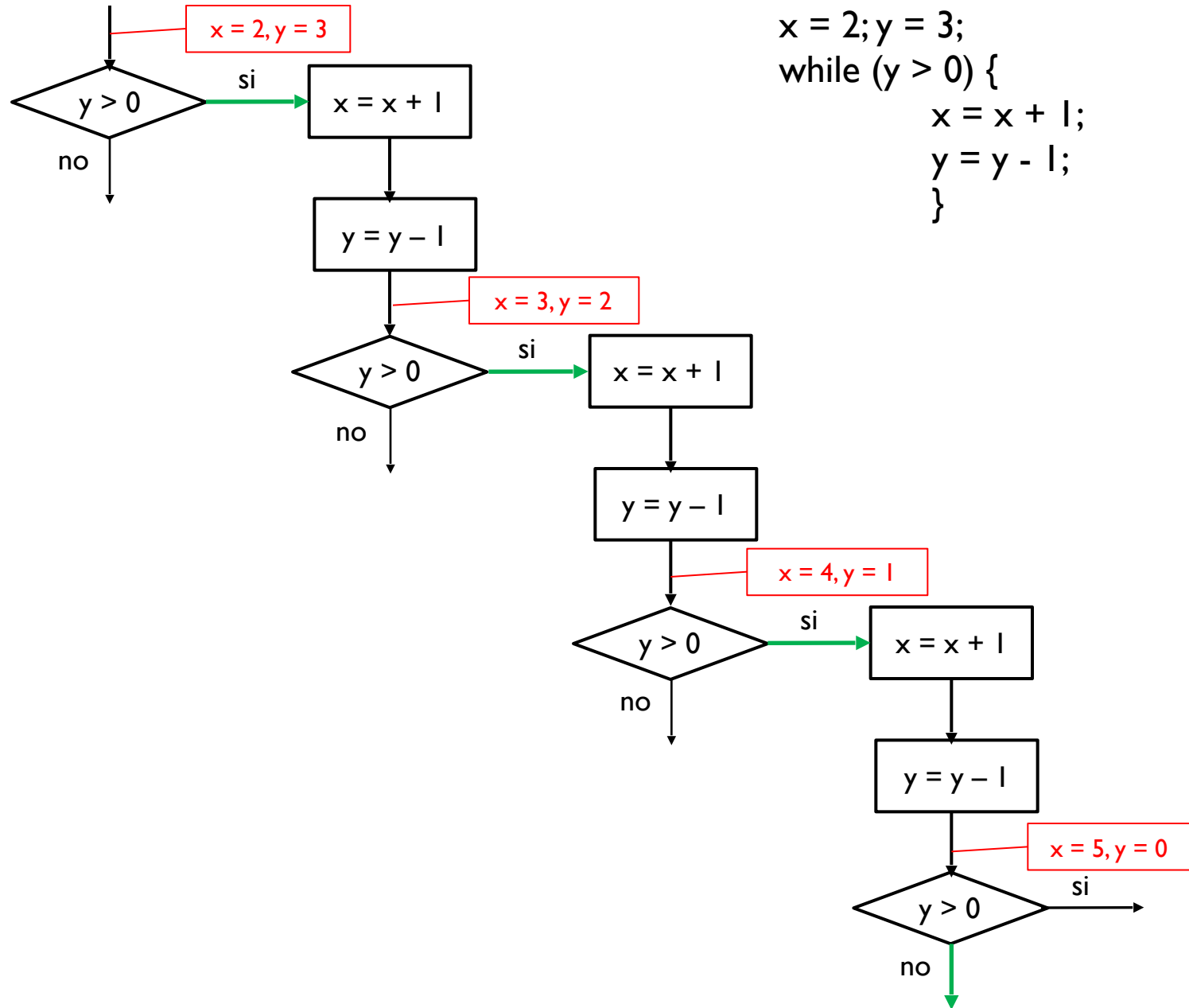




```
x = 2; y = 3;  
while (y > 0) {  
    x = x + 1;  
    y = y - 1;  
}
```



```
x = 2; y = 3;  
while (y > 0) {  
    x = x + 1;  
    y = y - 1;  
}
```



```
x = 2; y = 3;  
while (y > 0) {  
    x = x + 1;  
    y = y - 1;  
}
```

Istruzione while: esempio

```
x = 2; y = 3;  
while (y > 0) {  
    x = x + 1;  
    y = y - 1;  
}
```

x	y
-	-
2	-
2	3
3	2
4	1
5	0

Iterazione: istruzione do-while

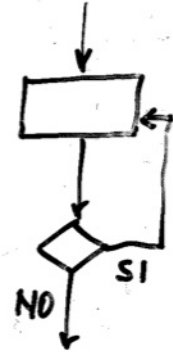
Istruzione do-while

Formato generale

do

<istruzione>

while (<espressione booleana>)



Esecuzione di un'istruzione do-while

1. si esegue l'istruzione;
2. si valuta l'espressione booleana:
 - (a) se la valutazione termina e si ottiene il valore true:
si torna al passo 1;
 - (b) se la valutazione termina e si ottiene il valore false:
si prosegue con la (eventuale) prossima istruzione del programma

Traduzione `while` \Leftrightarrow `do-while`

```
while (b) S;       $\equiv$       if (b)
                               do {
                                   S;
                               }
                               while (b);
```

```
do {
    S;
} while (b);       $\equiv$       S;
                               while (b)
                               S;
```

Istruzione for

(Savitch §4.1.3, pag. 142)

Formato generale

for (<inizializzazione del contatore>;<condizione>;<aggiornamento del contatore>)
 <istruzione>

Esecuzione di un ciclo for

- (1) si inizializza il contatore;
- (2) si valuta l'espressione booleana;
 - (a) se la valutazione termina e si ottiene il valore true:
 - si esegue <istruzione>;
 - si aggiorna il contatore;
 - si torna al passo (2)
 - (b) se la valutazione termina e si ottiene il valore false:
 - si prosegue con la (eventuale) prossima istruzione del programma

Istruzione for: semantica

Un'istruzione

for (I_1 ; E ; I_2) I_3

ha la stessa semantica dell'istruzione

```
 $I_1$ ;  
while ( $E$ ) {  
     $I_3$ ;  
     $I_2$ ;  
}
```