

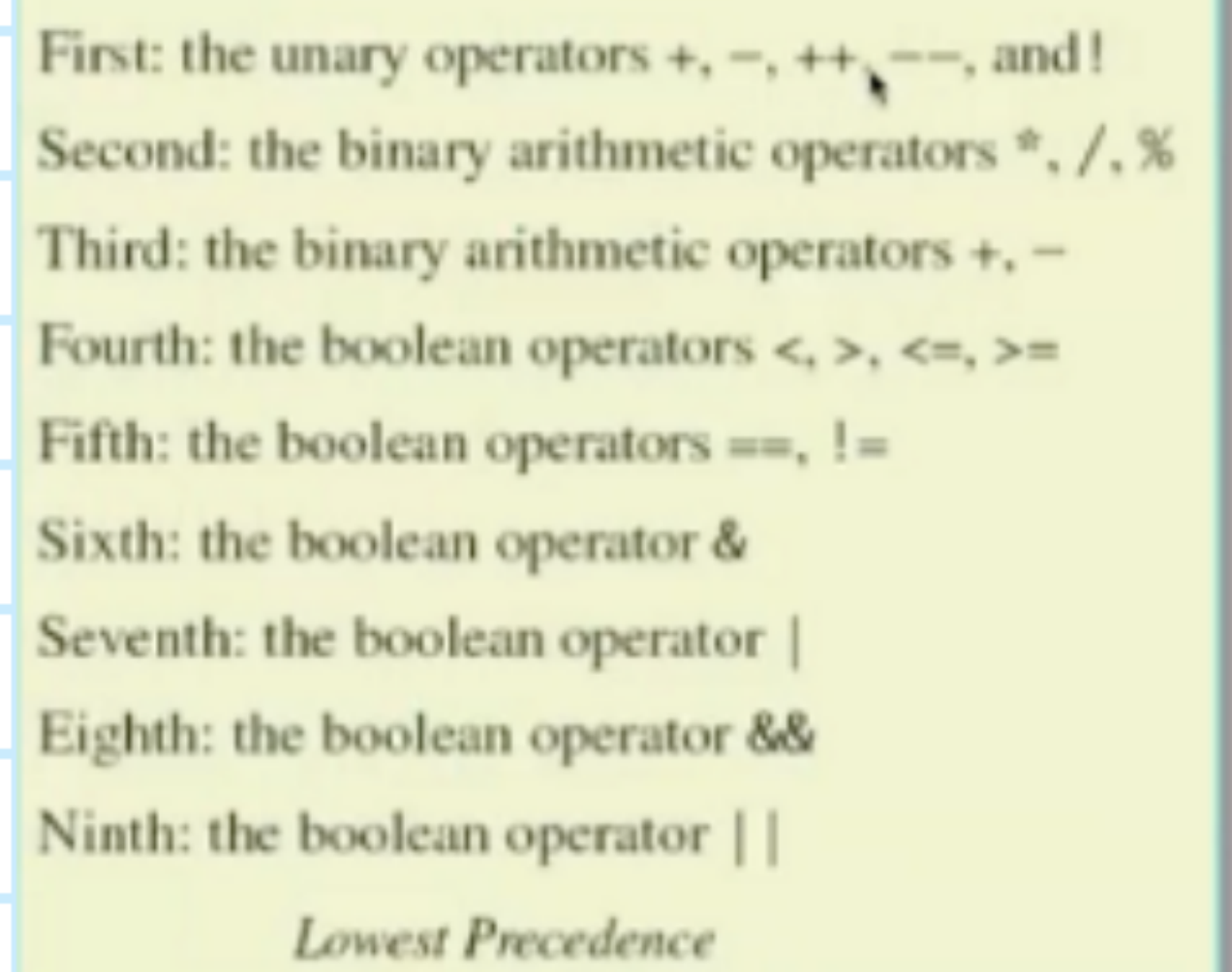
# Flusso Controllo

venerdì 29 ottobre 2021 01:55

-> Finora abbiamo fatto tutto sequenziale, ora aggiungiamo if (branching statement)

I cicli ripetono

Regole precedenza:



ShortCircuit / lazy

- Se abbiamo ||, e il primo è vero, il secondo non verrà calcolato
- Se abbiamo &&, e il primo è falso, il secondo non verrà calcolato

Questo può essere bypassato usando

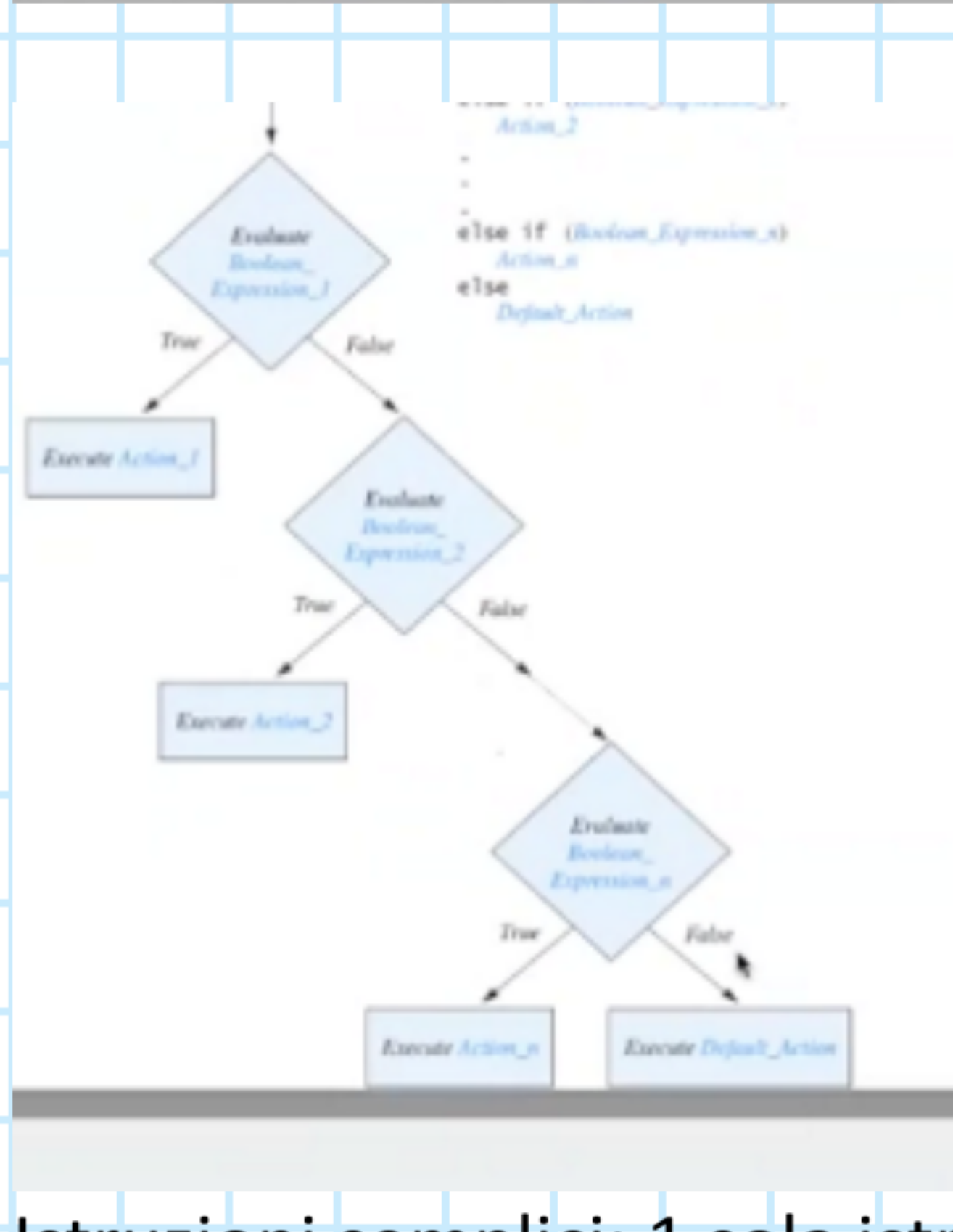
& e |

If else annidato -> Tanti if else dentro

Primo caso	Secondo caso
<pre>if (a &gt; b) {     if (c &gt; d)         e = f; } else     g = h;</pre>	<pre>if (a &gt; b)     if (c &gt; d)         e = f; else     g = h;</pre>

If else multiramo -> Tanti if stessa linea

```
if (espressione_booleana_1)
    azione_1;
else if (espressione_booleana_2)
    azione_2;
else if (espressione_booleana_3)
    azione_3;
else if ...
else
    azione_di_default;
```



Istruzioni semplici: 1 sola istruzione

Istruzioni complesse: 2+ istruzioni -> Necessarie {}

Ternario

```
int y = (x%2==0) ? ((x/2)*(x/2)) : (x+1)/2;

if (x%2==0)
    y = (x/2)*(x/2);
else
    y = (x+1)/2;
```

Espressioni booleane semplici: == > <

Complesse: && ||

System.exit(0) -> Esce

Per i valori booleani, dare nomi interrogativi

Switch è pog

La visibilità della variabili dipende da dove si trova.

O sono locali al metodo main, oppure if, oppure funzioni

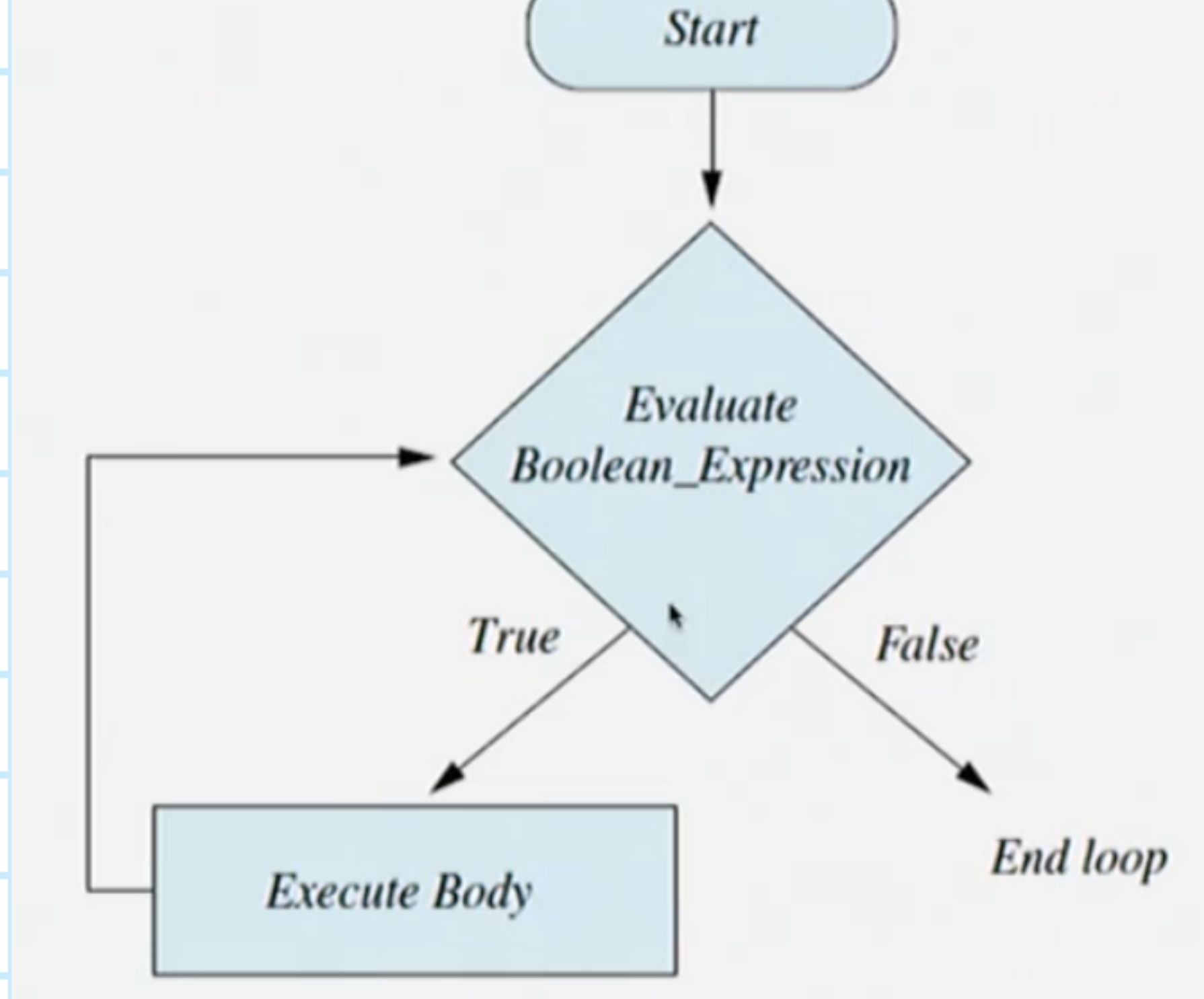
La porzione di programma che ripete un istruzione è

chiamata ciclo

\-> Possiede un body

\-> Ogni ripetizione del body è chiamata iterazione

\-> C'è una fine



LOL MI SONO ACCORTO CHE STAVO SEGUENDO LE

LEZIONI DEL TURNO 2, RIP

C'è da dire che spiega meglio

Math.random() -> [0, 1)

Tra 1 e 6: [1, 6]

(int)(Math.random()\*6)+1

For(inizializzazione, espressione, aggiornamento) {}

Int valore = 0;

While(valore > 0) {int n = valore; valore--; print(n)}

\-> Fino a che questo è vero

Ricorda: n è solo dentro ciclo, scope nel while

Per terminare:

- 1) Contatore -> Lanci dadi -> for
- 2) Richiesta utente -> Menu utente -> do while
- 3) Valore sentinella (quando viene trovato carattere # in un testo) -> while  
Public static final char sentinella = '#';  
-> Mi fermo quando ho trovato un carattere che cercavo
- 4) Variabile booleana

Si può avere un ciclo infinito con for tipo

For(;;) {}

Break -> Interrompe

Continue -> Prossima successione

Con i for è possibile fare multiple inizializzazioni

Terminare ciclo:

- Ask before iterating
- Count controlled
- Valore sentinella