Iniziamo a provare le condizioni.

Immagine che contiene testo, schermata, Carattere, bianco e nero

Il contenuto generato dall'IA potrebbe non essere corretto.

Le condizioni booleani o controllori del flusso è un concetto visto ieri all’interno degli elementi. Noi andiamo a strutturare una logica che si basa sempre su true e false, indifferentemente da come andiamo ad incapsulare le funzionalità, le logiche devono riportare true o false. Serve per ragionare sui controlli. L’esempio è nella slide sopra. L’espressione booleana è la condizione effettiva che va a valutare il true o false .

Immagine che contiene testo, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

Ogni tipo di controllo (uguaglianza, etc) riporterà alla macchina sempre una delle due condizioni, non potrà mai riportare altro e noi non vogliamo che lo riporti

Immagine che contiene testo, schermata, Carattere, bianco e nero

Il contenuto generato dall'IA potrebbe non essere corretto.

Le espressione booleane possono passare dalle variabili. Se vogliamo fare uguaglianza tra stringhe ci sono due elementi funzionali:

* Uppercase e lowercase, così che ogni input sia maiuscolo o minuscolo;
* Equals, cioè stringa1 e stringa2 che daranno o true o false (riga 3 esempio1).

quello sotto, è utilizzare il controllo dell’esempio 1, per andare a fare una scelta. Si vede qui il concetto di alterazione del flusso, anche detto struttura condizionale.

Immagine che contiene testo, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

La prima parte: una struttura condizionale è un elemento indifferentemente da quale esso sia, che si occupa di andare a scegliere il processo di elaborazione. La differenza tra questo e un elemento generico è che noi stiamo dando alla possibilità diretta all’elemento di andare a modificare come il flusso del codice funziona. Esempio, un blocco di codice può essere eseguito (quindi nessun obbligo), dall’altra parte siamo obbligati a strutturare dei controlli di due tipi:

* Iterazione (ciclo)
* Condizionale (scelta) -> ci concentreremo oggi su questo

Teorema di bohm-jacopini spiega il teorema di cui si basano le logiche della programmazione moderna.

Immagine che contiene testo, diagramma, schermata, design

Il contenuto generato dall'IA potrebbe non essere corretto.

Vediamo le condizioni. Noi partiamo da un punto di origine. Poi controllo (x>y per es), se la risposta è si eseguo un blocco di istruzioni, altrimenti non eseguo niente, o altrimenti l’else.

Immagine che contiene testo, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

Questo è un esempio di if. Andiamo prima alla teoria

Tutte le cose viste (and or e not, o == ,<= etc) vanno bene all’interno delle condizioni. È possibile ottimizzare procedure utilizzando anche meno condizioni possibili anche per eseguire diverse azioni per decisioni diverse.

L’if ha delle regole: (i blocchi condizionali possibili in java)

* Blocco if e l’if else;
* Blocco switch.