

# Orale

martedì 8 febbraio 2022 20:17

Se una domanda ha \* questo significa che non è una domanda che ho sentito nel mio girone di orali  
Però potrebbe venire richiesta/potrebbe utilizzarla per arricchire il vostro orale

## 1) Trasforma un ciclo while in un ciclo do while

La differenza fa il ciclo while e il do while è che il do while non controlla la prima esecuzione.  
Quindi, dobbiamo far sì che il ciclo while assicuri almeno 1 esecuzione.

Una veloce opzione è quella di aggiungere un valore booleano:

Boolean inizio = true;

```
While([NOSTRE OPERAZIONE BOOLEANE] || inizio) {  
    [LE NOSTRE OPERAZIONI]  
    Inizio = false;  
}
```

## 2) Trasforma il ciclo do while in while

Qui invece, il while prima di eseguire fa un controllo quindi, basta aggiungere un if:

```
If([NOSTRE OPERAZIONE BOOLEANE])  
    Do{}while()
```

## 3) Che cos'è il record di attivazione?

E' una zona di memoria nello stack che contiene tutte le informazioni

Necessarie per gestire correttamente l'esecuzione di un metodo invocato:

- o Parametri
- o Variabili locali della funzione
- o Indirizzo di rientro (dove poi il codice dovrà ritornare)
- o Valore di ritorno

Viene posto in cima allo stack e rimane fino a che la funzione è attiva.

Qua viene utilizzata la politica LIFO: last input first output.

## 4) Cos'è un algoritmo?

Gli algoritmi sono una sequenza non ambigua ed eseguibile autonomamente di istruzioni.

\-> Caratteristiche:

- Non ambiguo -> passi elementari
- Determinismo -> stessi input, stesso output
- Terminazione -> Deve finire

Inoltre, gli algoritmi devono:

- o Funzionare
- o Essere efficienti

Vengono scritti attraverso un linguaggio di programmazione.

## 5) Cos'è un algoritmo ricorsivo?

- o Un algoritmo ricorsivo è quando un algoritmo  
Contiene una versione ridotta dell'algoritmo completo.
- o Continua a richiamare se stesso e ritornare un valore fino a che una certa  
Condizione non è soddisfatta.
- o Un algoritmo ricorsivo è più facile da leggere però utilizza più memoria  
Ed è più lento.
- o Siccome crea tanti record di attivazioni, si può incorrere in un  
Stack overflow per via di una saturazione della memoria.

## 6) Variabili nello stack e nello heap

Stack:

- o Abbiamo tutte le variabili
- o Parte statica del nostro programma

Heap:

- o Abbiamo tutte le variabili che non sono delle primitive  
(classi, array, ...) -> Nello stack avremo un riferimento  
Alla cella di memoria del nostro oggetto
- o Parte dinamica

## 7) Cos'è un array?

Una sequenza di variabili omogenee, distinguibile a seconda della loro posizione

Nella sequenza (indice).

In particolare, è un oggetto viene dichiarato attraverso 'new'.

Ha la proprietà length che definisce la lunghezza dell'array e, la variabile di tipo array

E' contenuta nello stack ed è un puntatore verso l'head dove è contenuto l'oggetto array.

= -> Copia indirizzo

== -> Verifica se due array hanno lo stesso indirizzo di memoria

Nota:

L'array viene salvato sia nello stack che nell'heap.

Nello stack viene salvato l'indirizzo di memoria in cui si trova l'oggetto con tutti i vari elementi

E nell'heap si trova questo oggetto

Detto questo, quindi gli elementi di un array vengono salvati nell'heap e non nello stack.

Curiosità:

Le variabili: String ciao = "si"; String no = "si";

Hanno lo stesso indirizzo di memoria per via di un'ottimizzazione di java se però cambiassimo

String no = "no"; Non avrebbero più lo stesso indirizzo di memoria.

8) Che tipo di linguaggio è Java?

Compilato ed Interpretato.

Prima un qualunque file java viene compilato in bytecode,

Per la compilazione si esegue da linea di comando il comando `javac file.java`

E poi il bytecode viene interpretato in linguaggio macchina

Per l'interpretazione si esegue `java file`

Perché?

Siccome java cerca di prendere il pro sia di un linguaggio compilato che di uno interpretato

9) Che tipi puoi usare in un costrutto switch?

- Char
- Int
- String

10) Cosa succede se dimentichiamo un break dentro ad uno switch?

Si passa al case successivo.

Switch(a)

Case 1:

Break;

Case 2:

Case 3:

Break;

In questo caso, entrando dentro al case 2 si entra anche dentro al case 3 per via della mancanza del break.

Mentre lo stesso non succede per il case 1.

11) Come fare un confronto fra float?

Per via dell'imprecisione del float NON possiamo fare un semplice ==

Dobbiamo controllare questo float in un determinato intervallo.

Float val = 55.1;

Float delta = 0.001;

`(val - delta > val && val + delta < val)`

12) Cos'è un'espressione booleana?

E' un'espressione che, dati input qualsiasi, ritorna un valore booleano (true/false)

Operatori che ritornano valori booleani:

- &&
- ||
- >, >=
- <, <=
- ==
- & (and bit a bit) -> Non permette il "lazy circuit"
- | (or bit a bit) -> Non permette il "lazy circuit"

13) Cos'è il lazy circuit? \*

Il lazy circuit è quell'operazione di risparmio che viene utilizzata dall'operatore && e ||.

Praticamente, se con && il primo valore è falso non andremo a controllare il secondo operatore.

Se con il || il primo operatore è vero non andremo a controllare il secondo operatore.

Per evitare che questo avvenimento succeda possiamo usare & e |

14) Cos'è un cast?

Il cast è quell'operazione che ci permette di trasformare un qualsiasi tipo in un altro.

Esistono 2 tipologie di cast:

- Implicito: Avviene quando possiamo trasformare una variabile da un tipo ad un altro  
Senza però perdere precisione (int->long, int->float, float->double, short->int)  
Non c'è bisogno di specificare il tipo.

- Esplicito: Trasformare una variabile da un tipo all'altro con il rischio di Perdere precisione oppure di eventuali errori. (float->int, perdiamo la parte decimale)  
Dobbiamo specificare il tipo.

- 15) Come si calcola il numero che una lettera è rispetto all'alfabeto?  
Per farlo bisogna usare la tabella ascii.  
Ogni lettera ha un proprio valore in questa tabella, es, a -> 97  
E le lettere maiuscole sono diverse da quelle minuscole: A -> 65  
Prima di tutto dobbiamo controllare se stiamo lavorando con una lettera maiuscola oppure minuscola  
Char lettera = 'a';  
Boolean isMinuscola = lettera >= 'a' && lettera <= 'z'  
Se è minuscola noi dobbiamo sottrarre quella lettera per la 'a'  
Se è maiuscola invece dobbiamo farlo con la 'A'  
Int alfabeto = lettera - 'a';
- 16) Cos'è un paramametro formale/attuale?  
Il parametro forma è ciò che noi definiamo quando scriviamo un metodo, in particolare ciò che scriviamo Dentro le parentesi.  
Void funzione(int val, String ciao){}  
Int val, string ciao -> il nostro parametro formale  
Praticamente andiamo a definire ciò che ci aspetteremo che tale funzione riceverà.  
Mentre, il parametro attuale è ciò che effettivamente noi inseriamo quando chiamiamo tale funzione.  
Funzione(145, "si");
- 17) Qual è la differenza fra passaggio per parametro e per referenza? \*
- Quando noi facciamo un passaggio per parametro noi andiamo a inserire un valore primitivo come parametro  
Ed andiamo a fare letteralmente una copia di quel valore.  
Mentre, se noi non stiamo passando un tipo primitivo ma un oggetto (es, array), noi stiamo passando l'indirizzo di memoria.
- 18) Cos'è il Driver e lo Stub? \*
- Il driver viene utilizzato per verificare la giusta correttezza di un determinato metodo; andiamo a testarlo.  
Lo stub invece è una versione semplificata di quel metodo e viene utilizzato per temporaneamente fino a che Il metodo vero e proprio non viene codificato. Questo viene spesso usato quando si collabora in un grande progetto articolato.