RECAP CICLI

Esistono due tipi di cicli

* Ciclo determinato. Esegue un numero determinato e finito di cicli. Non è legata a nessun'altra condizione di controllo.
* Ciclo indeterminato. Esegue un numero indeterminato di cicli perché è legata a una condizione di controllo.

L'istruzione FOR crea un ciclo determinato ossia esegue un blocco di istruzioni per N volte.

La sintassi:

for ( inizio; condizione; incremento )

{ istruzioni; }

* Inizio. Il primo parametro inizializza la variabile contatore assegnandola a un valore iniziale ( es. i=0 )
* Condizione. Il secondo parametro fissa la condizione di controllo, in genere è legato alla variabile contatore ( es. i<10 ).
* Incremento. Il terzo parametro indica l'incremento della variabile contatore al termine di ogni ciclo ( i++ ).

L'istruzione WHILE crea un ciclo indeterminato.

Il blocco di istruzioni viene eseguito per un numero di volte indefinito a priori.

L'iterazione si interrompe soltanto quando l'espressione di controllo diventa falsa.

La sintassi:

while (condizione)

{ istruzioni; }

Nota. L'espressione di controllo è posta all'inizio del ciclo. Quindi il blocco di istruzioni potrebbe anche non essere eseguito nemmeno una volta

L'istruzione DO WHILE esegue il blocco di istruzioni per un numero indefinito di volte.

La struttura DO WHILE è simile alla struttura WHILE.

Tuttavia, in questo caso l'espressione di controllo è posta alla fine del ciclo. Quindi la struttura DO WHILE esegue il blocco di istruzioni almeno una volta o più.

La sintassi:

do{

istruzioni;

} while (condizione) ;

Se l'espressione di controllo è falsa, il blocco di istruzione viene comunque eseguito una volta.

Esercizio 1

Scrivere un programma JAVA *TabellineAritmetiche*che chiede in ingresso all’utente un numero intero da memorizzare nella variabile n e visualizza in uscita la tabellina aritmetica del numero stesso. Se ad esempio il numero inserito è 5 le righe visualizzate devono essere  
5×1=5  
5×2=10  
5×3=15  
5×4=20  
5×5=25  
5×6=30  
5×7=35  
5×8=40  
5×9=45  
5×10=50

Esercizio 2:

Scrivere un programma JAVA *Potenze*che chiede in ingresso all’utente un numero intero da memorizzare nella variabile n e visualizza in uscita le prime dieci potenze che hanno per base il numero stesso. Se ad esempio il numero inserito è 3 le righe visualizzate devono essere  
La potenza 0-esima di 3 e’ 1  
La potenza 1-esima di 3 e’ 3  
La potenza 2-esima di 3 e’ 9  
La potenza 3-esima di 3 e’ 27  
La potenza 4-esima di 3 e’ 81  
….

Esercizio 3:

Scrivere un programma TuttiPositiviPari che chiede all’utente di inserire una sequenza di interi (chiedendo prima quanti numeri voglia inserire) e poi, al termine dell’inserimento dell’intera sequenza, stampa "Tutti positivi e pari" se i numeri inseriti sono tutti positivi e pari, altrimenti stampa "NO".

Esercizio 4:

Scrivere un programma ContinuaFino che chiede all’utente di inserire una sequenza di interi. Il programma continua a chiedere all’utente di inserire valori fintanto che i valori inseriti soddisfano almeno una delle seguenti condizioni:

• è negativo e divisibile per 2

• è positivo e divisibile per 3

Al termine il programma stampa la somma di tutti i valori inseriti escluso quello ha violato le condizioni.

Esercizio 5:

Scrivere un programma MediaMultipliDiTre che chiede all’utente di inserire una sequenza di interi (chiedendo prima quanti numeri voglia inserire) e poi stampa la media di tutti i numeri inseriti che siano divisibili per tre. Per esempio, se si immettono i valori 5, 8, 9, 12, 7, 6 ,1 il risultato stampato dovrà essere 9.

Esercizio 6:

Scrivere un programma Contrario che chiede all’utente di inserire una stringa e la stampa al contrario. Per esempio, se si immette la stringa "Viva Java", il programma stampa "avaJ aviV"