

Programmazione orientata agli oggetti (OOP)

[Home](#)

Esercizi For

Esercizio 1

Scrivere un programma che stampa i numeri da 1 a 100 su righe diverse.

Esercizio 2

Scrivere un programma che stampa i numeri da 100 a 1 su righe diverse.

Esercizio 3

Scrivere un programma interattivo che chiede all'utente di inserire un numero n e stampa i primi n numeri dispari (1, 3, 5, 7, 9, ecc.)

Esercizio 4

Scrivere un programma che visualizza tutti i numeri compresi tra 20 e 30, uno per riga, in ordine decrescente a partire dal più grande (30, 29, 28, ..., 21, 20).

Esercizio 5

Scrivere un programma che chiede all'utente due numeri interi strettamente positivi a e b e stampa tutti i multipli di a minori di b. Ad esempio, se a=7 e b=50, il programma deve stampare: 7, 14, 21, 28, 35, 42, 49. Usare il seguente algoritmo (procedimento): passare in rassegna tutti i numeri tra 1 e b e usare un'istruzione if per decidere se un numero è divisibile per a.

Esercizio 6

Trovare una soluzione alternativa all'esercizio 5 che non faccia uso dell'istruzione if.

Esercizio 7

Scrivere un programma che chiede all'utente un intero n, e stampa la somma dei primi n numeri interi. Ad esempio, se l'utente inserisce 5, il programma deve stampare 15 (ottenuto come 1+2+3+4+5).

Esercizio 8

Scrivere un programma che chiede all'utente una sequenza di numeri decimali, uno alla volta. Il programma termina quando l'utente fornisce un numero negativo, visualizzando la media dei valori non negativi precedentemente inseriti.

Esercizio 9

Scrivere un programma che legge un intero maggiore di 1 e stampa tutti i suoi divisori (diversi da 1). Nella fase di lettura del valore, se l'utente immette numeri minori di 2 bisogna continuare a ripetere la richiesta del valore. Questo programma richiede l'uso di due cicli (non annidati) [T1, esercizio 6.4.5].

Esercizio 10

Scrivere un programma che chiede all'utente un numero decimale x e un numero intero n , e calcola x^n visualizzando il risultato su schermo. Ad esempio:

- inserendo 2.0 e -1, il programma deve stampare 0.5.
- inserendo 2.3 e 0, il programma deve stampare 1.0.
- inserendo 3.14 e 2, il programma deve stampare 9.8596.

Esercizio 11

Scrivere un programma che genera dapprima un numero casuale intero compreso tra 0 e 999 e poi chiede ripetutamente all'utente di indovinare quale numero sia stato generato. Per ogni tentativo errato, il programma dirà come suggerimento se il numero inserito è troppo basso o troppo alto, tenendo un conto dei tentativi effettuati. Quando l'utente indovina il numero, il programma stamperà "complimenti: hai indovinato in x tentativi", dove x è numero di tentativi che sono stati effettuati.

Suggerimento: per assegnare un intero casuale tra 0 e $k-1$ a una variabile p di tipo `int`, usare il comando `p = (int)(Math.random()*k);`.

Esercizio 12

Scrivere un programma che chiede all'utente un intero n e stampa tutti i numeri primi minori di n . Si ricordi che un numero è primo se è divisibile solo per 1 e per se stesso.

Esercizio 13

Scrivere un programma (costituito dal solo metodo `main`) che chiede all'utente di inserire una stringa formata da soli caratteri minuscoli, e stampa il numero di lettere "a" contenute nella stringa. Ad esempio, se l'utente immette "ciao sono una stringa", il programma deve stampare 3, poiché la stringa immessa contiene 3 lettere "a".

Esercizio 14

Modificare il programma dell'esercizio 13 in modo che stampi il numero di vocali nella stringa immessa. Ad esempio, se l'utente immette "ciao sono una stringa", il programma deve stampare 9, poiché la stringa immessa contiene 9 vocali.

Esercizio 15

Scrivere un programma (costituito dal solo metodo `main`) che chiede all'utente di inserire due stringhe a e b , e dice se a appare come sottostringa di b o meno. Ad esempio:

- l'utente immette $a="burg"$ e $b="hamburger"$, il programma stampa "hamburger contiene burg"
- l'utente immette $a="re"$ e $b="prezzo"$, il programma stampa "prezzo contiene re"

- l'utente immette `a="casa"` e `b="ventosa"`, il programma stampa `"ventosa non contiene casa"`
 - Suggerimento: si usino `length`, `substring` e `equals`.
-

Realizzato con [Drupal](#)