

# Corso di Programmazione 1

## Sesta Esercitazione di Laboratorio

### Esercizio 1

Scrivere un programma Java che stampi a video gli elementi di *posizione pari* di una sequenza di numeri interi positivi inserita dall'utente. Il primo numero positivo avrà posizione 1, il secondo posizione 2, ecc.; i numeri negativi vengono ignorati, e non incrementano la posizione; la sequenza termina quando l'utente inserisce 0.

Il programma **non** deve usare array; costruirà invece la stringa di output man mano che l'utente inserisce i numeri. Inoltre, il programma deve essere composto da due metodi:

- il `main()`, in cui si leggono i numeri interi, rifiutando quelli negativi con un messaggio d'errore, e si mantiene la stringa che andrà stampata alla fine dell'esecuzione;
- il metodo `aggiornaStringa()`, che prende come argomento un numero intero positivo e la stringa attuale di output, e restituisce al chiamante la nuova stringa di output, con il numero aggiunto in fondo.

### Esercizio 2

Scrivere un programma Java formato da due classi: `Programma` e `Metodi`. La classe `Programma` contiene solamente il metodo `main()`, che mostra a video il seguente menù:

- a) somma di due numeri interi
- b) divisione intera e resto tra due numeri interi
- c) media tra due numeri double
- d) valore assoluto di un numero intero
- e) lunghezza della stringa che rappresenta un numero intero
- t) termina il programma

e legge una delle lettere a, b, c, d, e, t (indifferentemente in minuscolo o maiuscolo), rifiutando con un messaggio d'errore qualsiasi altro carattere. Determinata la funzione che l'utente vuole eseguire, legge uno o due numeri (interi o double, a seconda della funzione scelta) e richiama gli opportuni metodi della classe `Metodi` per calcolare il risultato (o risultati) da stampare a video.

La classe `Metodi`, pertanto, conterrà i seguenti metodi, di cui vanno specificati opportunamente i parametri formali e il tipo di dati restituito al chiamante:

- `sommaDueInteri()`: restituisce al chiamante la somma dei due numeri interi passati come argomenti;
- `divisioneIntera()`: restituisce al chiamante il valore della divisione intera tra due numeri interi passati come argomenti. Questo metodo va chiamato solamente se il divisore è diverso da 0;
- `restoDivisioneIntera()`: restituisce al chiamante il resto della divisione intera tra due numeri interi passati come argomenti. Anche in questo caso, il divisore dovrebbe essere diverso da 0;
- `mediaTraDueDouble()`: restituisce al chiamante la media aritmetica tra due numeri double passati come argomenti;
- `valoreAssolutoIntero()`: restituisce al chiamante il valore assoluto di un numero intero passato come argomento;
- `lunghezzaStringaIntero()`: restituisce al chiamante la lunghezza della stringa che rappresenta il numero intero passato come argomento.

### Esercizio 3

Scrivere un programma Java che, letti da tastiera una stringa e un carattere, richiama il metodo `cercaCarattere()` per stabilire se la stringa contiene il carattere specificato; in tal caso il metodo restituisce il valore booleano `true`, in caso contrario restituisce `false`.

Nel caso in cui il metodo `cercaCarattere()` abbia restituito `true`, invocare il metodo `rimuoviCarattere()` che, dati come argomenti la stringa e il carattere inseriti da tastiera, restituisce una nuova stringa in cui ogni occorrenza del carattere è stata rimossa. Stampare a video la stringa restituita dal metodo.

### Esercizio 4

Scrivere un programma Java che permetta all'utente di convertire temperature da Fahrenheit a Celsius o viceversa, fino a quando decide di uscire dal programma (gestire quindi un menù a 3 scelte: Fahrenheit → Celsius, Celsius → Fahrenheit, Uscita). Il programma dovrà contenere i seguenti metodi:

- `menu()`: stampa a video il menù per l'utente, e restituisce al chiamante la scelta effettuata;
- `convertiFinC()`: prende come argomento una temperatura espressa in gradi Fahrenheit, e restituisce al chiamante la temperatura corrispondente in gradi Celsius;
- `convertiCinF()`: prende come argomento una temperatura espressa in gradi Celsius, e restituisce al chiamante la temperatura corrispondente in gradi Fahrenheit.

Si ricorda che le formule di conversione da Fahrenheit a Celsius, e viceversa, sono le seguenti:

- $C = (F - 32) / 1,8$
- $F = 1,8 * C + 32$

### Esercizio 5

Scrivere un programma Java che permetta all'utente di convertire numeri interi positivi da decimale a binario o viceversa, fino a quando decide di uscire dal programma (gestire quindi un menù a 3 scelte: Decimale → Binario, Binario → Decimale, Uscita). Il programma dovrà contenere i seguenti metodi:

- `menu()`: stampa a video il menù per l'utente, e restituisce al chiamante la scelta effettuata;
- `convertiDecInBin()`: prende come argomento un numero intero positivo (prima di chiamare il metodo, verificare che il numero sia positivo) espresso in decimale (es.: 243), e restituisce al chiamante la stringa che corrisponde al numero specificato espresso in binario;
- `convertiBinInDec()`: prende come argomento un numero intero positivo espresso in binario (notare che **non** stiamo usando la notazione in complemento a 2, quindi se il bit più significativo è 1 considerare il numero come positivo) e restituisce al chiamante l'intero corrispondente.

### Esercizio 6

Scrivere un programma Java che, lette da tastiera due stringhe `genoma` e `proteina`, entrambe formate dalle lettere A, C, G, T (che indicano rispettivamente *Adenina*, *Citosina*, *Guanina* e *Timina*, le quattro basi azotate che compongono il DNA) stampi a video tutte le occorrenze della stringa `proteina` all'interno della stringa `genoma`, usando i metodi `substring()` e `indexOf()` della classe `String` (a differenza della scorsa esercitazione, in cui si chiedeva di non usarli). Tuttavia, anziché invocare i metodi della classe `String`, fare finta che non si possano usare e reimplementarli (nella stessa classe del metodo `main()`).

**Ad esempio**, se `genoma = ACCAGTCAGTGCAATC` e `proteina = AGT`, il programma stamperà a video:

La sequenza AGT compare in ACCAGTCAGTGCAATC nelle  
seguenti posizioni: 4, 8

Si noti che la prima posizione ha indice 1.

**Altro esempio:** se `genoma = AAAAAAA` e `proteina = AAA`, il programma stamperà a video:  
La sequenza AAA compare in AAAAAAA nelle seguenti  
posizioni: 1, 2, 3, 4, 5