

- 1) In Java è possibile trasformare una String in intero attraverso l'istruzione Integer.parseInt()

Es.:

```
String numeroString = "5";  
int numeroInt = Integer.parseInt(numeroString);
```

Scrivete un programma che stampi la somma degli argomenti che l'utente ha inserito a riga di comando supponendo che l'utente digiti esclusivamente numeri interi. Es. Se l'utente digita

```
java Sommatore 1 2 3 4 5
```

il programma stamperà un messaggio del tipo:

```
La somma è 15
```

➤ Cosa succede se uno degli argomenti non è un intero?

- 2) Scrivere un semplice programma (una classe con il metodo main) che svolga le seguenti operazioni aritmetiche correttamente, scegliendo accuratamente i tipi di dati da utilizzare per immagazzinare i risultati di esse.

- Una divisione (usare il simbolo /) tra due interi  $a = 5$ , e  $b = 3$ . Immagazzinare il risultato in una variabile  $r1$ , scegliendo il tipo di dato adeguato.
- Una moltiplicazione (usare il simbolo \*) tra un char  $c = 'a'$ , ed uno short  $s = 5000$ . Immagazzinare il risultato in una variabile  $r2$ , scegliendo il tipo di dato adeguato.
- Una somma (usare il simbolo +) tra un int  $i = 6$  ed un float  $f = 3.14F$ . Immagazzinare il risultato in una variabile  $r3$ , scegliendo il tipo di dato adeguato.
- Una sottrazione (usare il simbolo -) tra  $r1$ ,  $r2$  e  $r3$ . Immagazzinare il risultato in una variabile  $r4$ , scegliendo il tipo di dato adeguato.
- Verificare la correttezza delle operazioni stampandone i risultati parziali ed il risultato finale. Tenere presente la promozione automatica nelle espressioni e utilizzare il casting propriamente.

- 3) Copiare, salvare e compilare la seguente classe:

```
public class NumeroIntero {  
    public int numeroIntero;  
  
    public void stampaNumero() {  
        System.out.println(numeroIntero);  
    }  
}
```

Questa classe definisce il concetto di numero intero come oggetto. In essa vengono dichiarati una variabile intera ed un metodo che stamperà la variabile stessa.

Scrivere, compilare ed eseguire una nuova classe (contenente ovviamente un metodo main()) che:

- istanzierà almeno due oggetti dalla classe NumeroIntero;
- cambierà il valore delle relative variabili d'istanza e testerà la veridicità delle avvenute assegnazioni, sfruttando il metodo stampaNumero();
- aggiungerà un costruttore alla classe NumeroIntero che inizializzi la variabile d'istanza.
- Se istanziamo un oggetto della classe NumeroIntero, senza assegnare un nuovo valore alla variabile numeroIntero, quanto varrà quest'ultima?

- 4) Creare una classe `Quadrato` che dichiari una variabile d'istanza intera `lato`. Creare un metodo pubblico che si chiami `perimetro()` che ritorni il perimetro del quadrato, e un metodo pubblico `area()` che ritorni l'area del quadrato.
- Creare una classe `TestQuadrato` contenente un metodo `main()` che istanzi un oggetto di tipo `Quadrato`, con `lato` di valore 5 (con una istruzione simile alla seguente: `nomeOggetto.lato = 5;`). Stampare poi il perimetro e l'area dell'oggetto appena creato.
  - Si crei un costruttore nella classe `Quadrato` che prenda in input il valore della variabile `lato`. Fatto questo si compili la classe `Quadrato`.
  - Ricompilare la classe `TestQuadrato` e interpretare l'errore.
  - Modificare il codice della classe `TestQuadrato` in modo tale che compili e sia eseguita correttamente.
  - Nella classe `Quadrato` sostituire il valore 4 usato per calcolare il perimetro con una costante d'istanza `NUMERO_LATI`.
- 5) Creare una classe `Rettangolo` equivalente alla classe `Quadrato`. Prima di codificare la classe decidere che specifiche deve avere questa classe (variabili e metodi).
- Si crei una classe `TestRettangolo` contenente un metodo `main()` che testi la classe `Rettangolo`, equivalentemente a come fatto nell'esercizio precedente. Istanziare almeno due rettangoli diversi.

**NOTE PER COMPILAZIONE E TEST A RIGA DI COMANDO IN AMBIENTE LINUX:**

|   |                                     |
|---|-------------------------------------|
| <b><code>javac nomeClasse.java</code></b> | <i>compila e genera il bytecode</i> |
| <b><code>java nomeClasse</code></b>       | <i>esegue il bytecode sulla JVM</i> |