

Esercizi aggiuntivi

2. Overloading dell'operatore +

Abbiamo visto che l'operatore + può indicare la somma tra interi (e, come vedremo più avanti, tra altri tipi numerici) e la concatenazione di stringhe. La semantica viene stabilita in fase di compilazione. In altre parole, essa è nota leggendo il testo del programma.

Esercizio 2.1

Considerate il codice seguente:

```
import prog.io.ConsoleInputManager;
import prog.io.ConsoleOutputManager;

class TestInteri {
    public static void main(String[] a) {
        ConsoleInputManager in = new ConsoleInputManager();
        ConsoleOutputManager out = new ConsoleOutputManager();

        int x = in.readInt("Inserire un numero intero ");
        int y = in.readInt("Inserire un altro numero intero ");

        int z = x + y;

        out.println("Messaggio 1: La somma vale " + z);
        out.println("Messaggio 2: La somma vale " + x + y);
        out.print("Messaggio 3: La somma vale ");
        out.println(x + y);

    }
}
```

Cercate di capire (solo leggendo il codice, senza farlo “girare” su una macchina!) cosa viene visualizzato dal programma. Fate poi compilare ed eseguire il programma e confrontate il risultato ottenuto con quanto vi aspettavate.

1. Spiegate il comportamento che, apparentemente, può sembrare anomalo, evidenziando le differenze tra le istruzioni che forniscono sul monitor le tre righe di messaggio.
2. Cosa vi aspettate succeda sostituendo la seconda chiamata di `println` con la seguente?

```
        out.println("Messaggio 2: La somma vale " + (x + y));
```

Cosa cambia? Dopo avere risposto, fate compilare ed eseguire il codice modificato per verificare la risposta.

Esercizio 2.2 (*Operatore + e riferimenti*)

Considerate il codice seguente:

```
import prog.io.ConsoleOutputManager;
import prog.utili.Frazione;

class Test {
    public static void main(String[] a) {
        ConsoleOutputManager out = new ConsoleOutputManager();

        Frazione f = new Frazione(1,2);
        Frazione g = null;

        out.println("Prima frazione " + f);
        out.println("Seconda frazione " + g);

        out.println("Prima frazione " + f.toString());
        out.println("Seconda frazione " + g.toString());

    }
}
```

1. Spiegate perché gli effetti prodotti dalla seconda e dalla quarta chiamata del metodo `println` sono differenti
2. Eliminate le ultime due chiamate di `println` e provate ad aggiungere una delle seguenti istruzioni
 - (a) `out.println(f + " " + g);`
 - (b) `out.println(f + g);`
 - (c) `out.println(f.toString() + g.toString());`

Cosa succede nei vari casi? Spiegate i comportamenti ottenuti.