

```
// Esempio di programmazione funzionale con Java

/**
 * Per ciascuna terna presente in un array di tre elementi verificare se essa
 * costituisce un triangolo. Restituire la lista dei perimetri per le terne che
 * costituiscono triangoli.
 *
 * Per la lista di interi formare le terne e verificare se rappresentano triangoli
 */
public class TerneArr {
    // Lista di terne come array di tre elementi
    private final List< int[] > terne = List.of(new int[] { 1, 2, 3 }, new int[]
        new int[] { 3, 4, 5 }, new int[] { 5, 2, 3 });

    // Lista di numeri da cui formare le terne
    private final List< Integer > listaNum = List.of(2, 2, 4, 6, 3, 6, 3, 3, 4, 5);

    /**
     * Trova le terne nelle quali ciascun numero è minore della somma degli altri
     * due
     */
    private List< int[] > trovaTriangoli(final List< int[] > listaTerne) {
        return listaTerne.stream()
            .filter(t -> t[0] < t[1] + t[2])
            .filter(t -> t[1] < t[0] + t[2])
            .filter(t -> t[2] < t[0] + t[1])
            .collect(Collectors.toList());
    }

    /** Calcola la somma della terna di lati */
    private List< Integer > calcolaPerimetri(final List< int[] > listaTerne) {
        return listaTerne.stream()
            .map(t -> t[0] + t[1] + t[2])
            .collect(Collectors.toList());
    }

    /** Forma array di tre elementi dati da tre numeri consecutivi della lista */
    private List< int[] > formaTerne() {
        final List< int[] > result = new ArrayList< >();
        for (int i = 0; i < listaNum.size() - 2; i++)
            result.add(new int[] { listaNum.get(i), listaNum.get(i + 1), listaNum.get(i + 2) });
        return result;
    }

    private void verificaCalcolaStampa(final List< int[] > ter) {
        ter.forEach(e -> System.out.print(e[0] + " " + e[1] + " " + e[2] + "\t"));
        final List< int[] > triang = trovaTriangoli(ter);

        System.out.println("\nTriangoli");
        triang.forEach(lati -> System.out.print(lati[0] + " " + lati[1] + " " + lati[2] + " " + lati[0] + lati[1] + lati[2] + "\n"));
    }
}
```

```

        System.out.println("\nPerimetri");
        final List< Integer > perim = calcolaPerimetri(triang);
        perim.forEach(p -> System.out.print(p + "\t"));
    }

    public static void main(final String[] args) {
        final TerneArr verif = new TerneArr();

        System.out.println("* Prima lista *");
        verif.verificaCalcolaStampa(verif.terne);

        System.out.println("\n* Seconda lista *");
        verif.verificaCalcolaStampa(verif.formaTerne());
        System.out.println();
    }
}

```

#### Output dell'esecuzione

```

* Prima lista *
1 2 3   3 2 3   3 3 3   3 4 5   5 2 3
Triangoli
3 2 3   3 3 3   3 4 5
Perimetri
8       9       12
* Seconda lista *
2 2 4   2 4 6   4 6 3   6 3 6   3 6 3   6 3 3   3 3 4   3 4 5
Triangoli
4 6 3   6 3 6   3 3 4   3 4 5
Perimetri
13      15      10      12

```