

```
// Esempio di programmazione funzionale con Java

import java.util.Arrays;
import java.util.List;
import java.util.Optional;

/**
 * Figura rappresenta un quadrilatero, e tiene la misura dei lati e degli angoli
 */
public class Figura {
    private final Integer[] lato = new Integer[4];
    private final Integer[] angolo = new Integer[4];
    private int i = 0;

    public Figura(final int l0, final int l1, final int l2, final int l3, final i
        lato[0] = l0;
        lato[1] = l1;
        lato[2] = l2;
        lato[3] = l3;
        angolo[0] = 360 - a1 - a2 - a3;
        angolo[1] = a1;
        angolo[2] = a2;
        angolo[3] = a3;
    }

    public int getLato() {
        if (i < 3)
            i++;
        else
            i = 0;
        return lato[i];
    }

    public int getAngolo() {
        if (i < 3)
            i++;
        else
            i = 0;
        return angolo[i];
    }

    public int getPerimetro() {
        return lato[0] + lato[1] + lato[2] + lato[3];
    }

    public int getLatoMinore() {
        final List< Integer > l = Arrays.asList(lato);
        final Optional< Integer > vmin = l.stream().min(Integer::compare);
        if (vmin.isPresent())
            return vmin.get();
        return 0;
    }
}
```

```
}

public boolean isQuadrato() {
    for (int i = 1; i < 3; i++)
        if (lato[i] != lato[0])
            return false;
    for (int i = 1; i < 3; i++)
        if (angolo[i] != angolo[0])
            return false;
    return true;
}
}

import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;

public class MainFigura {
    private static List< Figura > flista = List.of(new Figura(6, 3, 6, 3, 90, 90,
        new Figura(5, 5, 5, 5, 90, 90, 90));

    public static void main(String[] args) {
        // trova il lato minore fra tutte le figure
        Optional< Integer > minLato = flista.stream()
            .map(f -> f.getLatoMinore())
            .min(Integer::compare);

        if (minLato.isPresent())
            System.out.println("Lato minore: " + minLato.get());

        // trova i quadrati
        List< Figura > quadr = flista.stream()
            .filter(f -> f.isQuadrato())
            .collect(Collectors.toList());

        quadr.forEach(q ->
            System.out.println("Quadrato: " + q.getLato() + " " + q.getLato() + " "

        // trova somma perimetri
        int sumPerim = flista.stream()
            .map(f -> f.getPerimetro())
            .reduce(0, Integer::sum);

        System.out.println("Somma perimetri: " + sumPerim);
    }
}
```

Output dell'esecuzione

```
Lato minore: 3  
Quadrato: 5 5 5 5  
Somma perimetri: 56
```