

```

public class Navigazione {

    private static double udm = 1.0;           // unità di misura
    private static double diametroRuote = 0.055; // diametro delle ruote

    /**
     * @param x1 ascissa del punto di partenza
     * @param y1 ordinata del punto di partenza
     * @param x2 ascissa del punto di arrivo
     * @param y2 ordinata del punto di arrivo
     * @return distanza fra il punto di partenza e il punto di arrivo
     */
    public static double distanza(double x1, double y1, double x2, double y2) {
        return Math.sqrt(Math.pow(x1 - x2, 2.0) + Math.pow(y1 - y2, 2.0));
    }

    /**
     * @param nGiri numero di giri delle ruote
     * @return spazio percorso
     */
    public static double percorso(int nGiri) {
        return (Math.PI * diametroRuote * nGiri) / udm;
    }

    /**
     * @param udmValue valore dell'unità di misura da impostare
     */
    public static void setUdm(double udmValue) {
        if (udmValue > 0.0) {
            udm = udmValue;
        } else {
            System.out.println("valore non valido: udm = " + udmValue);
        }
    }

    /**
     * @return unità di misura impostata
     */
    public static double getUdm() {
        return udm;
    }
}



---



public class TestNavigazione {

    public static void main(String[] args) {

        System.out.println(Navigazione.distanza(2.0, 4.0, 3.0, 6.0));
        System.out.println(Navigazione.percorso(10));

        System.out.println("passo ai millimetri");
        Navigazione.setUdm(0.001);

        System.out.println(Navigazione.distanza(2000.0, 4000.0, 3000.0, 6000.0));
        System.out.println(Navigazione.percorso(10));

        System.out.println("passo a -3.0");
        Navigazione.setUdm(-3.0);
    }
}

```