

Retta con lettura da tastiera

Nota l'equazione di una retta $y = m \cdot x + q$, progettare un algoritmo che, impostati i valori di m , di q e del valore minimo di x , calcoli N punti equispaziati della retta, noto il passo di discretizzazione dx . Si desidera visualizzare i risultati in forma tabellare: per ogni punto deve essere visualizzato su video il valore di x e il corrispondente valore di y . L'algoritmo deve essere realizzato con **costrutto iterativo**.

Implementare con uso di linguaggio Java¹, volendo leggere da tastiera con uso di **oggetto di tipo Scanner**: proporre un **metodo leggiD()** per leggere con **controllo degli input**, verificando che il valore digitato sia un numero reale con la **virgola** come separatore decimale.

Listing in Java

```
/**
 * RettaLeggi.java
 *
 * RettaLeggi application
 *
 * @author 3INF
 * @version 1.00
 */
import java.util.Scanner; // dalla versione Tiger è disponibile la classe Scanner

public class RettaLeggi {

    public static double leggiD(){ // gestendo eccezioni
        double d; // variabile locale
        System.out.print("\n[virgola come separatore decimale]: "); // messaggio amichevole

        Scanner sc= new Scanner (System.in); // creo nuovo oggetto di tipo Scanner di nome sc
        // collegato alla tastiera

        try {
            d= sc.nextDouble(); // chiedo all'oggetto sc di eseguire un suo metodo:
            // leggere un double
        }catch (Exception e) {
            System.out.println("Ridigita !!");
            return leggiD() ;
        }
        return d; // Valore restituito
    }

    public static void main (String [] args) {

        final int N = 3; // variabile imm modificabile o costante – numero di punti
        double x, dx, m, q, // letti da tastiera
            y;
        System.out.print("\nScegli il valore del coefficiente angolare ");
        m = leggiD();
        System.out.print("Scegli il valore del termine noto ");
        q = leggiD();
        System.out.print("Scegli il valore iniziale delle ascisse ");
        x = leggiD();
        System.out.print("Scegli il passo di discretizzazione [o.d.g." + 1.0/10 + "]: ");
        dx = leggiD();
    }
}
```

Utilizzo del valore restituito



¹ Per soluzione **modulare**: risorsa dedicata (dall'[analisi](#) all'[implementazione](#))

```

System.out.println("\nAscisse\t\tOrdinate\n");
for (int i = 0; i < N; i++) {
    y = m*x + q;
    System.out.println("\t" + x + "\t\t" + y + "\n");
    x = x + dx;
}
System.out.println("Termine\n");
} // fine main
} // fine class

```

```

General Output
-----Configuration: Leggere - JDK version 1.8
Scegli il valore del coefficiente angolare
[virgola come separatore decimale]: 5
Scegli il valore del termine noto
[virgola come separatore decimale]: 10
Scegli il valore iniziale delle ascisse
[virgola come separatore decimale]: 0
Scegli il passo di discretizzazione [o.d.g.0.1] :
[virgola come separatore decimale]: 0,1

Ascisse      Ordinate
    0.0      10.0
    0.1      10.5
    0.2      11.0

Termine

Process completed.

```

Testare i possibili errori di digitazione:

- numero con **punto** come separatore delle cifre decimali

```

General Output
-----Configuration: Leggere
Scegli il valore del coefficiente angolare
[virgola come separatore decimale]: 5.0
Ridigita !!

[virgola come separatore decimale]: |

```



- lettera dell'alfabeto

```

General Output
-----Configuration: Leggere
Scegli il valore del coefficiente angolare
[virgola come separatore decimale]: s
Ridigita !!

[virgola come separatore decimale]:

```

