

## 17. Scheda di lavoro (OOP: oggetto corrente this, attributi e metodi statici)

08/01/2020

Si propone una soluzione dell'esercizio della scheda precedente che utilizza:

- l'oggetto corrente this
- attributi statici (di classe)
- metodi statici (di classe)

```

1  package pkg4b_2019.pkg20;
2  public class Main {
3      public static void main(String[] args) {
4          Cliente a = new Cliente("Gaias", "Federico", "123456789" );
5          Cliente b = new Cliente("Pulina", "Antonio", "789123456" );
6          ContoCorrente ccl1 = new ContoCorrente (a);
7          ContoCorrente cc2 = new ContoCorrente (b);
8          ContoCorrente cc3 = new ContoCorrente (b);
9          System.out.println(ContoCorrente.Presentazione());
10         ccl1.versamento(73);
11         ccl1.prelievo(15);
12         ccl1.prelievo(20);
13         cc2.versamento(10);
14         cc3.versamento(30.6);
15         System.out.println(ccl1.restituisceDatiCC());
16         System.out.println(cc2.restituisceDatiCC());
17         System.out.println(cc3.restituisceDatiCC());
18     }
19 } //fine classe applicazione

```

**Output - 4B\_2019-20 (run) x**

```

run:
STAI UTILIZZANDO LA CLASSE ContoCorrente
Gaias Federico   tel:123456789   numeroCC:1   saldo:38.0   numero movimenti:3
Pulina Antonio   tel:789123456   numeroCC:2   saldo:10.0   numero movimenti:1
Pulina Antonio   tel:789123456   numeroCC:3   saldo:30.6   numero movimenti:1

```

```

1  package pkg4b_2019.pkg20;
2  public class Cliente {
3      private String cognome;
4      private String nome;
5      private String telefono;
6      public Cliente(String co, String no, String tel){
7          cognome = co;
8          nome = no;
9          telefono = tel;
10     }
11     public String restituisciDati() {
12         return cognome + " " + nome + "   tel: "+ telefono;
13     }
14 }
15 }

```

```
1 package pkg4b_2019.pkg20;
2 public class ContoCorrente {
3     static int contatore = 1; // static serve a condividere i dati
4     // fra più istanze della stessa classe
5     int numeroCC;
6     double saldo;
7     int movimenti;
8     Cliente titolare;
9
10    public ContoCorrente(Cliente titolare) {
11        this.numeroCC = contatore;
12        contatore++;
13        this.titolare = titolare;
14        saldo = 0;
15        movimenti = 0;
16    }
17
18    public int getNumeroCC() {
19        return numeroCC;
20    }
21
22    public double getSaldo() {
23        return saldo;
24    }
25
26    public int getMovimenti() {
27        return movimenti;
28    }
29
30    public Cliente getTitolare() {
31        return titolare;
32    }
33
34    String restituisciDatiCC() {
35        return titolare.restituisciDati() + " numeroCC:" + numeroCC +
36            " saldo:" + saldo + " numero movimenti:" + movimenti;
37    }
38
39    public void versamento(double x) {
40        saldo = saldo + x;
41        movimenti++;
42    }
43
44    public void prelievo(float x) { //inserire qui eventuale controllo debito
45        saldo = saldo - x;
46        movimenti++;
47    }
48
49    public static String Presentazione() { //metodo di classe
50        return "STAI UTILIZZANDO LA CLASSE ContoCorrente ";
51    }
52
53 } //fine classe ContoCorrente
```