

32. Scheda di lavoro (OOP: incapsulamento, attributi e metodi statici)

07/03/2019

Considera la classe **ContoCorrente** mostrata dal modello UML in figura

Movimenti è un attributo che viene incrementato ogni volta che l'intestatario effettua un prelievo oppure un versamento

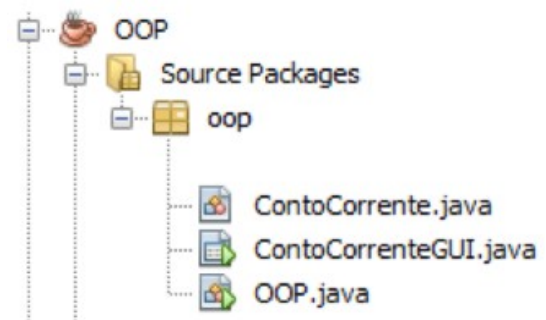
Prelievo e versamento sono due metodi che ricevono come parametro l'entità della variazione da effettuare sul conto e aggiornano il valore del saldo (per il prelievo attenzione: non è ammesso andare in debito, eventualmente si blocca l'operazione).

ContoCorrente
numeroConto intestatario saldo movimenti
Costruttori getNumeroConto getIntestatario getSaldo Prelievo Versamento

OOP: incapsulamento

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

```
package oop; //file oop.java
public class JavaApplication1 {
    // esempio utilizzo classe ContoCorrente
    public static void main(String[] args) {
        String nome = "leonardo";
        double sal = 300;
        ContoCorrente mio = new ContoCorrente(nome, sal);
        ContoCorrente tuo = new ContoCorrente("pippo",200);
        mio.versamento(73);
        mio.prelievo(15);
        mio.prelievo(15);
        String tmp;
        tmp= ContoCorrente.Presentazione(); //chiamata metodo di classe
        System.out.println(tmp);
        System.out.println("numero conto --> " + mio.getNumeroConto());
        System.out.println("intestatario --> " + mio.getIntestatario());
        System.out.println("movimenti --> " + mio.getMovimenti());
        System.out.println("saldo --> " + mio.getSaldo());
        System.out.println("-----");
        System.out.println("numero conto --> " + tuo.getNumeroConto());
        System.out.println("intestatario --> " + tuo.getIntestatario());
        System.out.println("movimenti --> " + tuo.getMovimenti());
        System.out.println("saldo --> " + tuo.getSaldo()); }
} // fine class
```



```
Output - OOP (run) x
run:
numero conto --> 1
intestatario --> leonardo
movimenti --> 3
saldo --> 343.0
-----
numero conto --> 2
intestatario --> pippo
movimenti --> 0
saldo --> 200.0
```

```
package oop; //file ContoCorrente.java
public class ContoCorrente {
    static int contatore = 1; //static serve a condividere i dati fra più istanze della stessa classe
    private int numeroConto;
    private String intestatario;
    private double saldo;
    private int movimenti;
```

```
public ContoCorrente( String intestatario, double saldo) {
    this.numeroConto = contatore;
    contatore ++;
    this.intestatario = intestatario;
    this.saldo = saldo;
    movimenti = 0;
}
public ContoCorrente(String intestatario) {
    this.numeroConto = contatore;
    contatore ++;
    this.intestatario = intestatario;
    saldo = 0;
    movimenti = 0;
}
public ContoCorrente() {
    this.numeroConto = contatore;
    contatore ++;
    this.intestatario = "sconosciuto";
    saldo = 0;
    movimenti = 0;
}

public int getNUmeroConto() {
    return numeroConto;
}
public String getIntestatario() {
    return intestatario;
}
public void setIntestatario(String nome) {
    this.intestatario = nome;
}
public double getSaldo() {
    return saldo;
}

public void versamento(int x) {
    saldo = saldo + x;
    movimenti++;
}
public void prelievo(int x) { //inserire qui eventuale controllo debito
    saldo = saldo - x;
    movimenti++;
}
public int getMovimenti() {
    return movimenti;
}

public static String Presentazione() //metodo di classe
{
    return "STAI UTILIZZANDO LA CLASSE ContoCorrente \n ";
}
}
```