

```

import java.io.*;
class ProgContoSpeciale {
    public static void main (String[] args) {

        InputStreamReader input =
            new InputStreamReader(System.in);
        BufferedReader tastiera =
            new BufferedReader(input);

        ContoCorrente ccOrd = new ContoCorrente ();
        ContoSpeciale conto = new ContoSpeciale ();
        Assegno versamento = new Assegno();

        double importo;

        System.out.println ( "versamento ---- ");
        importo = leggiDouble (tastiera , "assegno : ");
        versamento.importo = importo;
        //--- importo attributo pubblico dell'oggetto versamento

        conto.versa (versamento);
        ccOrd.versa (importo);

        conto.stampaSaldo ();
        ccOrd.stampaSaldo ();

        System.out.println ( "\nprelievo -----");
        importo = leggiDouble (tastiera , "prelievo: ");
        System.out.println ( "\nda ccSpeciale ---");
        conto.preleva(importo);
        conto.stampaSaldo ();
        System.out.println ( "\nda ccOrdinario --");
        ccOrd.preleva(importo);
        ccOrd.stampaSaldo ();
    }
    private static double leggiDouble
        (BufferedReader tast , String chiedi){
        double valore = 0;
        String leggiD;
        //-- controllo input su un singolo campo
        for (int k = 1; k <= 1 ; k++) {
            System.out.print ( "importo " + chiedi );
            try {
                leggiD = tast.readLine();
                valore = Double.valueOf(leggiD).doubleValue();
                if (valore <= 0) {
                    System.out.println ("---valore negativo o
nullo!\n");
                }
                k--;
            }
        }
        catch(Exception e){
            System.out.println ("---valore errato!\n");
            k--;
        }
    } //----- fine for per controllo input di un singolo campo
    return valore;
} //----- fine leggiDouble
}

```

Il programma alloca un oggetto **ccOrd** di classe **ContoCorrente** di tipo ordinario dal quale è possibile prelevare soldi anche in presenza di saldo insufficiente (situazione che nel mondo reale determina notevoli spese aggiuntive per lo scoperto) e alloca anche un oggetto **conto** di classe **ContoSpeciale** dal quale è possibile prelevare solo un importo limitato che non può comunque superare il saldo disponibile; sui **ContoSpeciale** si possono versare solo assegni bancari.

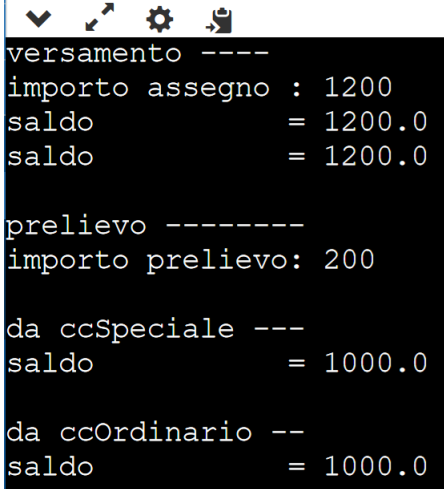
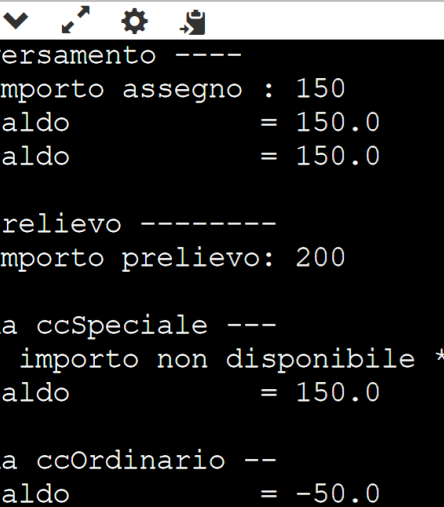
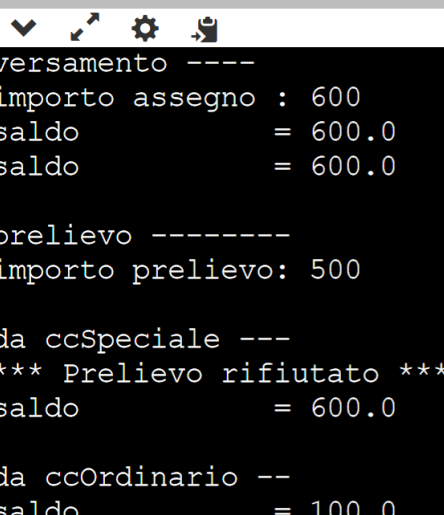
La classe **ContoCorrente** ha:

- un attributo privato **saldo** che la classe derivata **ContoSpeciale** (sottoclasse) può vedere solo se richiama il metodo **getSaldo()** (ereditato dalla superclasse)
- il metodo **stampaSaldo()** che è ereditato dalla sottoclasse **ContoSpeciale** (l'oggetto **conto** può richiamarlo)

La classe **ContoSpeciale** ha:

- il metodo **preleva** (double importo) che è ridefinito (**overriding**) e richiama i metodi **getSaldo()** e **preleva** (double importo) della superclasse **ContoCorrente**
- il metodo **versa** (**Assegno ass**) che è ridefinito (**overriding**): ha in input un oggetto di classe **Assegno** e richiama il metodo **versa** (double importo) della superclasse **ContoCorrente**

*NOTA: i metodi ridefiniti vanno richiamati premettendo **super**. (riferimento implicito alla **sopra**classe dell'oggetto di classe **ContoSpeciale**)*

classe base ContoCorrente (superclasse)	OUTPUT
<pre>class ContoCorrente { private double saldo; public void versa (double importo) { saldo += importo; } public void preleva (double importo) { saldo -= importo; } public double getSaldo () { return saldo; } public void stampaSaldo () { System.out.println("saldo \t= " + saldo); } }</pre>	 <pre> ▼ ↗ ⚙ 🗑 versamento ---- importo assegno : 1200 saldo = 1200.0 saldo = 1200.0 prelievo ----- importo prelievo: 200 da ccSpeciale --- saldo = 1000.0 da ccOrdinario -- saldo = 1000.0</pre>
<pre>class ContoSpeciale extends ContoCorrente { private final double LIMITE = 200.0; public void preleva (double importo) { if (importo <= LIMITE) if (importo <= getSaldo()) super.preleva(importo); else System.out.println("*importo non disponibile*"); else System.out.println("*** Prelievo rifiutato ***"); } public void versa (Assegno ass) { super.versa(ass.importo); } }</pre>	 <pre> ▼ ↗ ⚙ 🗑 versamento ---- importo assegno : 150 saldo = 150.0 saldo = 150.0 prelievo ----- importo prelievo: 200 da ccSpeciale --- * importo non disponibile * saldo = 150.0 da ccOrdinario -- saldo = -50.0</pre>
<pre>class Assegno { public double importo; }</pre>	 <pre> ▼ ↗ ⚙ 🗑 versamento ---- importo assegno : 600 saldo = 600.0 saldo = 600.0 prelievo ----- importo prelievo: 500 da ccSpeciale --- *** Prelievo rifiutato *** saldo = 600.0 da ccOrdinario -- saldo = 100.0</pre>
<p>soluzione ALTERNATIVA per l'attributo saldo</p> <pre>//--- in class ContoCorrente può essere definito : protected double saldo; //--- un attributo protected è visibile dalle sottoclassi della classe in cui è stato dichiarato, quindi : //--- in class ContoSpeciale nel metodo preleva (double importo) : if (importo <= saldo) //--- è visibile l'attributo saldo della sopraclasse saldo -= importo; //--- nel metodo versa (Assegno ass) : saldo += ass.importo; //--- e quindi può essere eliminato il metodo getSaldo()</pre>	