

Assegnamenti tra riferimenti a oggetti di superclasse/sottoclasse, metodi sovrascritti e late binding

Con riferimento alle classi `StudUniv` (superclasse) e `StudUnivFC` (sottoclasse) definite precedentemente, si considerino i seguenti oggetti:

```
StudUniv s1 = new StudUniv("Andrea", "Rossi", 134678);
StudUnivFC s2 = new StudUnivFC("Elena", "Bianchi",111789,3);
```

Il metodo `info`, presente nella superclasse, è sovrascritto nella sottoclasse e applicabile su entrambi gli oggetti `s1` ed `s2`:

```
System.out.println(s1.info());
System.out.println(s2.info());
```

dando luogo alla stampa delle seguenti stringhe:

```
Rossi Andrea (matr. 134678)
Bianchi Elena (matr. 111789) 3 anno fuori corso.
```

Se `s2` (riferimento di sottoclasse) è assegnato ad `s1` (riferimento di superclasse) e poi si cerca di applicare su `s1` un metodo definito nella sottoclasse:

```
s1 = s2;
String t = s1.leggiAnnoFC();
s1.aggAnnoFC();
```

si ha errore in fase di compilazione, in quanto tali metodi non appartengono alla classe a cui appartiene `s1`. Se invece si applica su `s1` un metodo presente nella superclasse che è sovrascritto nella sottoclasse, come `info`, per il meccanismo del *late binding*, viene eseguito il metodo sovrascritto nella sottoclasse, ovvero la classe a cui appartiene l'oggetto riferito da `s1`:

```
System.out.println(s1.info());
```

Viene infatti stampato:

```
Bianchi Elena (matr. 111789) 3 anno fuori corso.
```

Supponiamo di “perdere” il riferimento `s2` all'oggetto della sottoclasse e di voler assegnare `s1` ad un riferimento di sottoclasse:

```
s2 = null;
StudUnivFC s3 = s1;
```

Si ha errore in fase di compilazione in quanto i tipi di `s1` e di `s3` non sono “compatibili”. Tuttavia, il riferimento di superclasse `s1` riferisce un oggetto della sottoclasse, quindi può essere assegnato al riferimento di sottoclasse `s3` tramite l'uso del *cast* forzando il tipo di `s1`:

```
StudUnivFC s3 = (StudUnivFC) s1;
```

Per evitare errori nel caso in cui `s1` non riferisca effettivamente un oggetto della sottoclasse, è consigliabile usare `instanceof` per controllare che `s1` riferisca un oggetto della classe `StudUnivFC`, come segue:

```
StudUnivFC s3;
if (s1 instanceof StudUnivFC)
    s3 = (StudUnivFC) s1;
else s3 = null;
```

```
System.out.println(s3.info());
```

Nuovamente viene stampata la stringa:

```
Bianchi Elena (matr. 111789) 3 anno fuori corso.
```