

Programmazione Orientata agli Oggetti

Note sulla classe Razionale ed oltre

Libero Nigro

MCD col metodo di Euclide

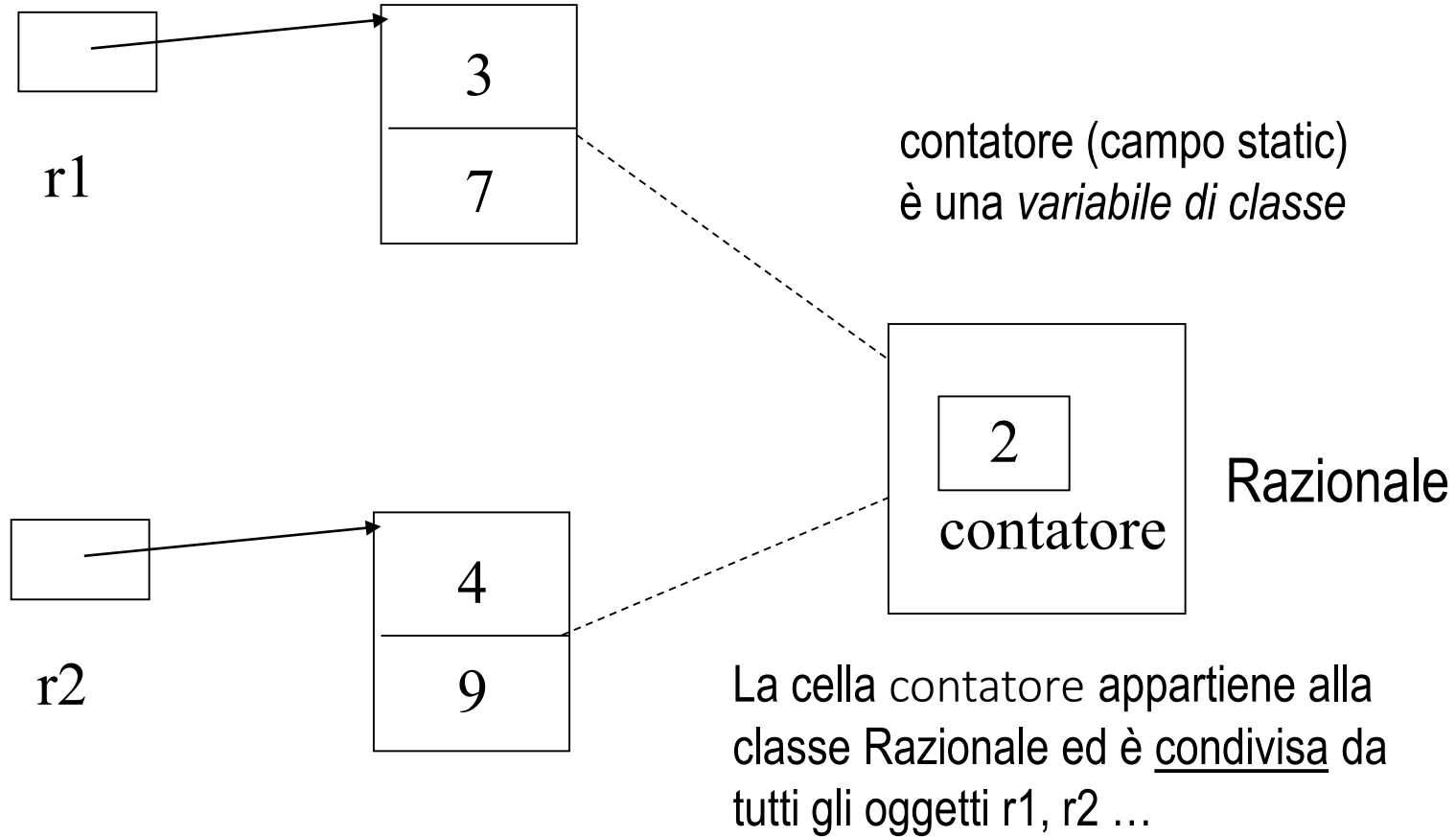
- Il metodo privato `mcd()` di `Razionale`, di seguito riprodotto, calcola il MCD per divisioni successive sino a che si incontra resto nullo

```
private int mcd( final int n, final int m ){//pre: n>0 && m>0 e non più modificabili
    int x=n, y=m;
    do{
        int r=x%y;
        x=y; y=r;
    }while( y!=0 );
    return x;
} //mcd
```

- Una versione alternativa di `mcd`, più sintetica, è costituita dal seguente *metodo ricorsivo* (si auto-invoca):

```
private int mcd( int x, int y ){
    if( y==0 ) return x;
    return mcd( y, x%y );
} //mcd
```

Uno scenario



Entità di istanza e entità di classe

- (non static) Variabili come **NUM** e **DEN** nella classe Razionale, e metodi come **add**, **sub** etc. costituiscono *entità (o attributi) d'istanza*, dal momento che hanno rilevanza per ogni singolo oggetto. Tali entità presuppongono l'uso di **this**
- (static) Variabili come **contatore** e metodi come **razionaliCreati**, costituiscono *entità di classe*, ossia che hanno rilevanza di classe, indipendentemente da qualsiasi istanza. Si rifletta che il loro significato sussiste anche quando nessun oggetto Razionale è stato ancora creato. Dunque **non** si basano su **this**

<i>Il metodo (colonna) può accedere a ?</i>	metodo non static (o di istanza)	metodo static (o di classe)
variabili o metodi non static (o di istanza)	Si	No
variabili o metodi static (o di classe)	Si	Si

Un metodo static non può invocare direttamente un metodo non static. Tuttavia, esso (es. il main) può creare un oggetto della classe, e su questo oggetto (this) invocare un metodo di istanza