

Esercizio 1.

Un numero razionale (o frazione) è un numero che può essere espresso come frazione di due numeri interi. Ad esempio $3/4$ e $7/100$ sono numeri razionali.

Si progetti una classe `Razionale` con due variabili d'istanza intere per rappresentare il numeratore ed il denominatore del numero razionale. (Nota: non è necessario che i numeri razionali siano ridotti, ad esempio $1/2$ e $2/4$ sono ugualmente accettabili).

Si progettino un costruttore ed i seguenti metodi della classe `Razionale`:

- `int getNumeratore()` : restituisce il numeratore;
- `int getDenominatore()` : restituisce il denominatore;
- `boolean maggioreDi(int n)` : restituisce `true` se il numero razionale è maggiore di `n`, `false` altrimenti;
- `Razionale somma(Razionale r)` : restituisce un nuovo numero razionale ottenuto sommando il numero `r`.

Si progetti una classe di test che contenga i seguenti metodi statici:

- `Razionale[] serie(int n)` : restituisce una array di numeri razionali i cui elementi sono $1/n$, $2/n$, $3/n$,... n/n ;
- `Razionale sommatoria(Razionale[] a)` : calcola la somma di tutti i numeri razionali presenti nell'array `a`.

Infine si scriva il metodo `main` nel quale si calcoli la sommatoria $1/100 + 2/100 + 3/100 + 4/100 + \dots + 100/100$ e si dica se è maggiore di 2 sfruttando i metodi statici sopra esposti.

Esercizio 2.

Si progettino una nuova classe `Razionale2`, sottoclasse di `Razionale`, ed una eccezione `EccezioneRazionale`.

Si progetti un costruttore della classe `Razionale2` affinché restituisca l'eccezione `EccezioneRazionale` se si cerca di costruire un numero razionale il cui denominatore è zero.

Si progetti un metodo per calcolare la divisione tra numeri razionali:

`Razionale2 divisione(Razionale2 r)`: effettua la divisione del numero razionale ed `r`, e solleva l'eccezione `EccezioneRazionale` se `r` è zero.

Esercizio 3.

Si dica cosa stampa il seguente programma, motivando la risposta.

```
public class A {
```

```
    private int n=1;

    public int calcola(int a) {
        return n+a;
    }
}

public class B extends A {

    public B() {
        super();
    }

    public int calcola(int a) {
        return super.calcola(a)+1;
    }
}
```

```
public class Test {

    public static void main(String[] args) {
        A[] a = new A[2];
        a[0] = new A();
        a[1] = new B();

        System.out.println(a[0].calcola(10));
        System.out.println(a[1].calcola(10));
    }
}
```