



package

utilizzo classi

- Java contiene molte classi predefinite che sono raggruppate in ***categorie di classi*** e prendono il nome di ***package***
- l'insieme di questi package è chiamato Java ***API*** (*Java Application Programming Interface*)
- per specificare le classi necessarie per compilare un programma Java viene utilizzata la dichiarazione import
 - **import <package>.<Classe>;**
- *esempio* per utilizzare la classe Scanner del package java.util si usa la seguente dichiarazione
 - **import java.util.Scanner;**
- per utilizzare tutte le classi del package java.util si usa la dichiarazione
 - **import java.util.*;**

- il package java.math contiene due classi dedicate alla cosiddetta "*aritmetica a precisione arbitraria*":
BigInteger e **BigDecimal**
- entrambe servono per gestire numeri la cui grandezza o precisione supera i limiti imposti dai tipi standard di Java: int, long, double, float, ecc.
- **java.math.BigInteger**
 - serve per rappresentare numeri interi immutabili di grandezza arbitraria; il massimo numero di cifre memorizzabili è **Integer.MAX_VALUE**
- **java.math.BigDecimal**
 - serve per rappresentare numeri decimali immutabili di precisione arbitraria

confronto tra int e BigInteger

A. Ferrari

```
int intero;  
BigInteger grandeIntero;  
intero = 2000000000;  
grandeIntero = BigInteger.valueOf(intero);  
System.out.println ("intero = "+intero+"; grandeIntero =  
"+grandeIntero);  
System.out.println ("Raddoppio...");  
intero *= 2;  
grandeIntero = grandeIntero.multiply(new BigInteger("2"));  
System.out.println ("intero = "+intero+"; grandeIntero =  
"+grandeIntero);
```

il risultato è il seguente:

```
intero = 2000000000; grandeIntero = 2000000000  
Raddoppio...  
intero = -294967296; grandeIntero = 4000000000
```

```
public static int fattoriale(int intero) {  
    int risultato = 1;  
    for(int i=2; i<=intero; i++) {  
        risultato *= i;  
    }  
    return risultato;  
}
```

```
public static BigInteger fattoriale(BigInteger grandeIntero) {  
    BigInteger risultato = new BigInteger("1");  
    for(BigInteger i = new BigInteger("2");  
        i.compareTo(grandeIntero)<=0;  
        i=i.add(new BigInteger("1"))  
    {  
        risultato = risultato.multiply(i);  
    }  
    return risultato;  
}
```

```
intero = 100;
grandeIntero = BigInteger.valueOf(intero);
System.out.println ("intero = "+intero+";
                    grandeIntero = "+grandeIntero);
intero = fattoriale(intero);
grandeIntero = fattoriale(grandeIntero);
System.out.println ("Fattoriale...");
System.out.println ("intero = "+intero+";
                    grandeIntero = "+grandeIntero);

--- risultato ---
intero = 100; grandeIntero = 100
Fattoriale...
intero = 0; grandeIntero =
9332621544394415268169923885626670049071596826438162146
8592963895217599993229915608941463976156518286253697920
8272237582511852109168640000000000000000000000000000
```

- realizzare un metodo statico
`int potenza(int b, int e)`
che restituisce come risultato il valore di **b** elevato a **e**
(*si suppone che **e** sia positivo*)
- testare il metodo prima con valori "piccoli" poi con valori "grandi"
- realizzare un metodo statico
`BigInteger potenza(BigInteger b, BigInteger e)`
che restituisce come risultato il valore di **b** elevato a **e**
(*si suppone che **e** sia positivo*)
- testare il metodo prima con valori "piccoli" poi con valori "grandi"