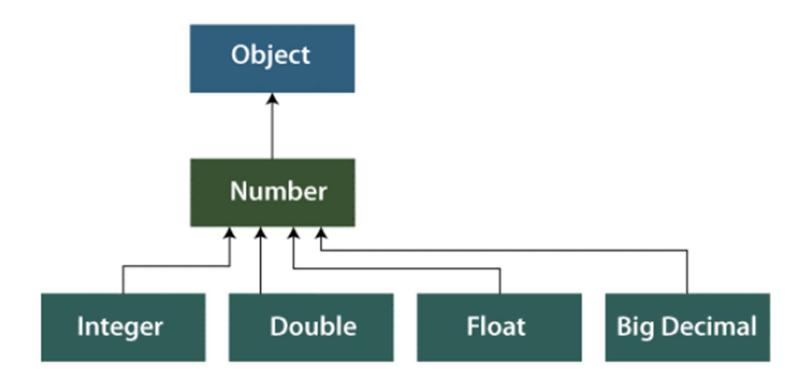


Module 6: Classe BigDecimal

"Classes BigDecimal"



O modo mais direto de criar um BigDecimal é:

BigDecimal x = new BigDecimal("123.45");

•

A classe **BigDecimal** faz parte do pacote **java.math**.

Usada para realizar aritmética decimal precisa e lidar com números com precisão arbitrária.

Ela é comumente usada em aplicações que requerem alta precisão e controle sobre o comportamento de arredondamento, como aplicações financeiras, conversões de moeda e outros cálculos que envolvem números decimais.

Nunca use um tipo **float** ou **double** para um valor exato, como valores monetários.

Usando BigDecimal com double ou long;

```
double d = 123.45;
BigDecimal x = new BigDecimal(d);
```

Usando constantes BigDecimal;

```
BigDecimal zero = BigDecimal.ZERO; // represents 0
BigDecimal one = BigDecimal.ONE; // represents 1
BigDecimal ten = BigDecimal.TEN; // represents 10
```

Os objetos BigDecimal são imutáveis.

Qualquer operação neles retornará um novo objeto BigDecimal com o resultado da operação, e o objeto BigDecimal original não será modificado.

Usando constantes BigDecimal;

```
BigDecimal zero = BigDecimal.ZERO; // representa 0
BigDecimal one = BigDecimal.ONE; // representa 1
BigDecimal ten = BigDecimal.TEN; // representa 10
```

Métodos BigDecimal

| Method | What it does |
|------------|---|
| add() | Adds one BigDecimal object value to another |
| subtract() | Subtracts one BigDecimal object value from another. |
| multiply() | Multiplies the value of one BigDecimal object by another. |
| abs() | Returns the absolute value of the given BigDecimal |
| pow(n) | Returns the BigDecimal to the power of n |

```
BigDecimal money1 = new BigDecimal("100.26");
BigDecimal money2 = new BigDecimal("59.75");
BigDecimal result;

// Calculate Sum
result = money1.add(money2);

// Calculate difference
result = money1.subtract(money2)
```

BigDecimal roudingMode

| | Result o | Result of rounding input to one digit with the given rounding mode | | | | | | | |
|--------------|----------|--|---------|-------|---------|-----------|-----------|--|--|
| Input Number | UP | DOWN | CEILING | FLOOR | HALF_UP | HALF_DOWN | HALF_EVEN | | |
| 5.5 | 6 | 5 | 6 | 5 | 6 | 5 | 6 | | |
| 2.5 | 3 | 2 | 3 | 2 | 3 | 2 | 2 | | |
| 1.6 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | | |
| 1.1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | | |
| 1.0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| -1.0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | | |
| -1.1 | -2 | -1 | -1 | -2 | -1 | -1 | -1 | | |
| -1.6 | -2 | -1 | -1 | -2 | -2 | -2 | -2 | | |
| -2.5 | -3 | -2 | -2 | -3 | -3 | -2 | -2 | | |
| -5.5 | -6 | -5 | -5 | -6 | -6 | -5 | -6 | | |

https://docs.oracle.com/javase/7/docs/api/java/math/RoundingMode.html

https://kevinsguides.com/guides/code/java/javaintro/05bigdecimal

Questions and Comments

 What questions or comments do you have?



