Theory: Infinity and Not a Number

3207 users solved this topic. Latest completion was about 2 hours ago.

9 minutes 5 / 11 problems solved

Start practicing

Java provides three special values for both floating-point types (`float` and `double`): `+Infinity`, `-Infinity` and `NaN`. The values appear in different cases and can be directly assigned to variables of these types.

- `Double.POSITIVE_INFINITY` is a constant that is greater than any number.

```
1  double posInf = Double.POSITIVE_INFINITY;  // +Infinity
2
3
double anotherPosInf = +1 / 0.0;          // it's +Infinity, not an exception
4
5  double posInfAgain = anotherPosInf + 100; // +Infinity again
```

In Java, when we write an expression with the division by the real zero `0.0`, no errors occur. We also can divide a real value by `0`.

- `Double.NEGATIVE_INFINITY` is a constant that is less than any number.

```
1  double negInf = Double.NEGATIVE_INFINITY; // -Infinity
2
3  double negInfAgain = negInf * 100;       // The result is -Infinity
4
5  double anotherNegInf = -1.002 / 0.0;     // It is also -Infinity
```

There are no errors in this case, either.

In general, the operations sum and product performed on an infinity value and a regular value return infinity with the same sign. For the operations subtract, remainder, and division, the result also depends on the order of operands.

- `Double.NaN` is a special constant that represents an undetermined value (such as `0 / 0`). It does not equal any floating-point or integer number. `NaN` is an acronym for **Not a Number**.

```
1  double nan = Double.NaN;         // the NaN constant
2
3  double anotherNan = 0.0 / 0.0;   // it's the NaN, not an exception
```

Also, this value occurs after some operations on infinity values.

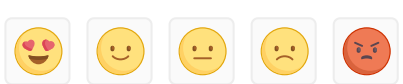
```
1
double nan = Double.NEGATIVE_INFINITY + Double.POSITIVE_INFINITY;     // NaN
2
3
double nanToo = Double.POSITIVE_INFINITY - Double.POSITIVE_INFINITY;  // Also NaN
4
5
double notANan = Double.POSITIVE_INFINITY + Double.POSITIVE_INFINITY; // it's +Inf
inity!
```

Any arithmetic operation with `NaN` produces `NaN` as the result. Actually, the results of arithmetic operations do not contradict the common sense.

The `float` type has the same special values: `Float.POSITIVE_INFINITY`, `Float.NEGATIVE_INFINITY`, `Float.NaN`. The results of arithmetic operations are the same as the ones we discussed above.

Report a typo

249 users liked this theory. 7 didn't like it. What about you?



Current topic:
Infinity and Not a Number
Topic depends on:
Stage 5
Floating-point types
Table of contents:
↑ Infinity and Not a Number
Feedback & Comments

Start practicing

Comments (1) Hints (0) Useful links (0) Show discussion