

Integral quantity data types

[Technology](#), [Development](#)



INTEGRAL DATA TYPES IN JAVA Prepared by Presented to On Maximum and Minimum values for the INT data type Table1 outlines the minimum and maximum values that an Integer can store in Java Language (TutorialPoints, 2014):

Table 1 Integer Data Types

Data Type

Description

Maximum Value

Minimum Value

INT

It is 32 bit data type, which is used to store integer values.

2, 147, 483, 647

- 2, 147, 483, 648

2. The values look a bit odd in that they are not a nice rounded value like 1, 000, 000. Yet, these limits are not arbitrary. How are these limits determined?

Yes, Java programming language provides some rounding modes, which determine how a certain operation gives back the result in limited numbers of digits as compared to the number of digits that are returned by results. It can be achieved through `MathContext` precession setting. We can use this precession setting to specify the limits of any number that we want to round in digits (Oracle Corporation, 2014).

3. Give some examples where Javas limit on the maximum-sized integer it can represent poses a limitation and give some examples where we need to represent integer quantities outside of the range available in Java.

Given below are some of the examples of both scenarios (Coffey, 2011):

In case, when we need numbers which are greater than arbitrary range (less than 1×10^{307} and greater than 1×10^{307}).

In cryptography where we need to represent integer quantities outside of the range available in Java. For instance, for cryptographic algorithm such as RSA algorithm.

In case, when we need to represent decimal numbers precisely.

4. Is there a way in Java to represent integers bigger than what ints allow?

Why would Java provide two data types to represent integral quantities?

Yes, we can represent integers bigger than what int allows. Basically, integers have limit up to 2, 147, 483, 647, and if we want to show number greater than this we have two options. We can use long or BigInteger. In this scenario, the limit for long is 9, 223, 372, 036, 854, 775, 807. For storing space and increasing the time of execution of program Java provides two data types (TutorialPoints, 2014; Oracle Corporation, 2014).

5. Is there a way in Java for representing arbitrarily large integers? Why is not this the default? In other words, why not have just one representation for arbitrarily large integral quantities?

Yes, for representing arbitrarily large integers we can use two other options: BigInteger and BigDecimal. However, it is not default because we write small programs in which our integer values can easily store and take less space.

Hence, if BigDecimal or BigInteger becomes default it will require more space and the program execution will require additional time. Due to this reason there is not a single representation of integral quantities

(TutorialPoints, 2014; Oracle Corporation, 2014).

References

Coffey, N., 2011. BigDecimal and BigInteger: calculations with arbitrary precision in Java. [Online]

Available at: http://www.javamex.com/tutorials/math/BigDecimal_BigInteger.shtml

[Accessed 22 10 2014].

Oracle Corporation, 2014. Class BigDecimal. [Online]

Available at: <http://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html>

[Accessed 22 10 2014].

TutorialPoints, 2014. Java Basic Data Types. [Online]

Available at: http://www.tutorialspoint.com/java/java_basic_datatypes.htm

[Accessed 22 10 2014].