



Java

Большие числа

Лекция #5

Пустовалова О.Г.
доцент. каф. мат.мод.
ИММИКН ЮФУ

Содержание

- Для чего нужны Большие числа
- Неизменяемость больших чисел
- BigInteger из пакета java.math
- BigDecimal из пакета java.math
- Примеры

Большие числа

<https://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html>

<https://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html>

Большие числа

Если для решения задачи недостаточно точности основных типов, чтобы представить целые и вещественные числа, то можно обратиться к классам **BigInteger** и **BigDecimal** из пакета **java.math**.

Эти классы предназначены для выполнения действий с числами, состоящими из **произвольного количества цифр**.

В классах **BigInteger** и **BigDecimal** реализуются арифметические операции **произвольной точности** соответственно для целых и вещественных чисел.

Большие числа

Размер чисел BigInteger и BigDecimal ограничен памятью компьютера

Используются большие числа для вычислений с крайне высокими требованиями к точности:

- Финансовые расчеты
- ПО для самолетов и ракет
- Медицинское оборудование

Миллион	10^6
Миллиард	10^9
Триллион	10^{12}
Квадриллион	10^{15}
Квинтиллион	10^{18}
Секстиллион	10^{21}
Септиллион	10^{24}
Октиллион	10^{27}
Нониллион	10^{30}
Дециллион	10^{33}

Большие числа. BigInteger

Для преобразования обычного числа в число с произвольной точностью (называемое также *большим числом*) служит статический метод `valueOf ()`:

```
BigInteger a = BigInteger.valueOf(100);
```

```
BigInteger a = BigInteger.valueOf(2147483647);
```

```
BigInteger b = BigInteger.valueOf(2147483641);
```

```
//a = a + b; // так нельзя
```

```
a = a.multiply(b);
```

```
System.out.println(a);
```



```
4611686001247518727
```


**Основные методы
класса BigInteger из пакета java.math**

Большие числа. Основные методы класса `BigInteger`

- `BigInteger add (BigInteger other)`: возвращает сумму двух чисел
- `BigInteger subtract (BigInteger other)`: возвращает разность двух чисел
- `BigInteger multiply (BigInteger other)`: возвращает произведение двух чисел
- `BigInteger divide (BigInteger other)`: возвращает частное двух чисел
- `BigInteger mod (BigInteger other)`: возвращает остаток от целочисленного деления двух чисел

Большие числа. Основные методы класса BigInteger

- BigInteger **sqrt** (): возвращает квадратный корень числа
- int **compareTo** (BigInteger other): сравнивает два числа. Возвращает -1, если текущий объект меньше числа other, 1 - если текущий объект больше и 0 - если числа равны
- static BigInteger **valueOf** (long x): возвращает объект BigInteger, значение которого равно числу, переданному в качестве параметра

Большие числа. Основные методы класса `BigInteger`

- `int intValue ()`: конвертирует объект `BigInteger` в объект `int`
- `byte byteValue ()`: преобразует объект `BigInteger` в `byte`
- `short shortValue ()`: преобразует объект `BigInteger` в `short`
- `long longValue ()`: преобразует объект `BigInteger` в `long`

**Примеры использования методов
класса BigInteger из пакета java.math**

Большие числа. Константы. Количество бит. Сдвиги

```
BigInteger o1 = BigInteger.ONE; // 1  
BigInteger o2 = BigInteger.TEN; // 10  
BigInteger o3 = BigInteger.ZERO; // 0
```

```
// количество бит bitCount()  
System.out.println(BigInteger.valueOf(255).bitCount()); // 8
```

```
// Сдвиг влево  
System.out.println(BigInteger.valueOf(2).shiftLeft(2)); // 8  
  
// Сдвиг вправо  
System.out.println(BigInteger.valueOf(64).shiftRight(2)); // 16
```

Большие числа. Сравнение

// сравнение. возвращает ноль, если равны

```
System.out.println(BigInteger.valueOf(255).compareTo(BigInteger.valueOf(255)));
```

// возвращает 1, если первое больше второго

```
System.out.println(BigInteger.valueOf(256).compareTo(BigInteger.valueOf(255)));
```

// возвращает -1, если первое меньше второго

```
System.out.println(BigInteger.valueOf(255).compareTo(BigInteger.valueOf(256)));
```

**Основные методы
класса `BigDecimal` из пакета
`java.math`**

Большие числа. Основные методы класса `BigDecimal`

- `BigDecimal add (BigDecimal other)`: возвращает сумму двух чисел
- `BigDecimal subtract (BigDecimal other)`: возвращает разность двух чисел
- `BigDecimal multiply (BigDecimal other)`: возвращает произведение двух чисел
- `BigDecimal divide (BigDecimal other)`: возвращает частное двух чисел
- `BigDecimal divide (BigDecimal other, RoundingMode mode)`: результат деления двух чисел, округленное в соответствии с режимом `mode`

Большие числа. Основные методы класса `BigDecimal`

- `int compareTo(BigDecimal other)`: сравнивает два числа. Возвращает -1, если текущий объект меньше числа `other`, 1 - если текущий объект больше и 0 - если числа равны
- `static BigDecimal valueOf(double x)`: возвращает объект `BigDecimal`, значение которого равно числу, переданному в качестве параметра
- `double doubleValue()`: преобразует объект `BigDecimal` в `double`
- `float floatValue()`: преобразует объект `BigDecimal` в `float`

Большие числа. BigDecimal

```
BigDecimal c = BigDecimal.valueOf(2325.06);
```

```
BigDecimal d = BigDecimal.valueOf(215.06);
```

```
c = c.subtract(d.multiply(BigDecimal.valueOf(2.1)));
```

```
System.out.println(c);
```



1873.434

```
double y = c.doubleValue();
```

```
System.out.println(y);
```



1873.434



Пример

Большие числа. Пример

```
import java.math.*;

public class Main {
    public static void main(String[] args) {

        long a, b, c;
        BigInteger d, e, f;

        a = 9223372036854775807L;
        b = a;
        c = a * b;

        d = BigInteger.valueOf(a);
        e = BigInteger.valueOf(b);
        f = d.multiply(e);

        System.out.println("(long) " + a + " * " + b + " = " + c);
        System.out.println("(BigInteger) " + d.toString() + " * " + e.toString() +
" = " + f.toString());
    }
}
```

Большие числа. Пример. Результат

(long) 9223372036854775807 * 9223372036854775807 = **1**

(BigInteger) 9223372036854775807 * 9223372036854775807 =

85070591730234615847396907784232501249



Спасибо за внимание!