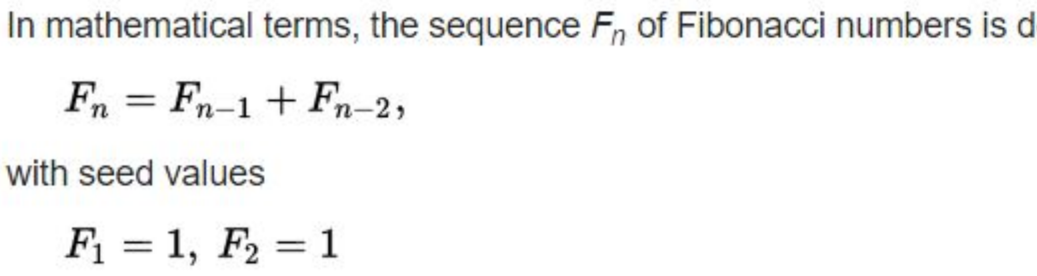
Primitive types and operations

Kamill Gusmanov
@GusmanovKamill

Fibonacci number

1, 1, 2, 3, 5, 8, 13, 21…

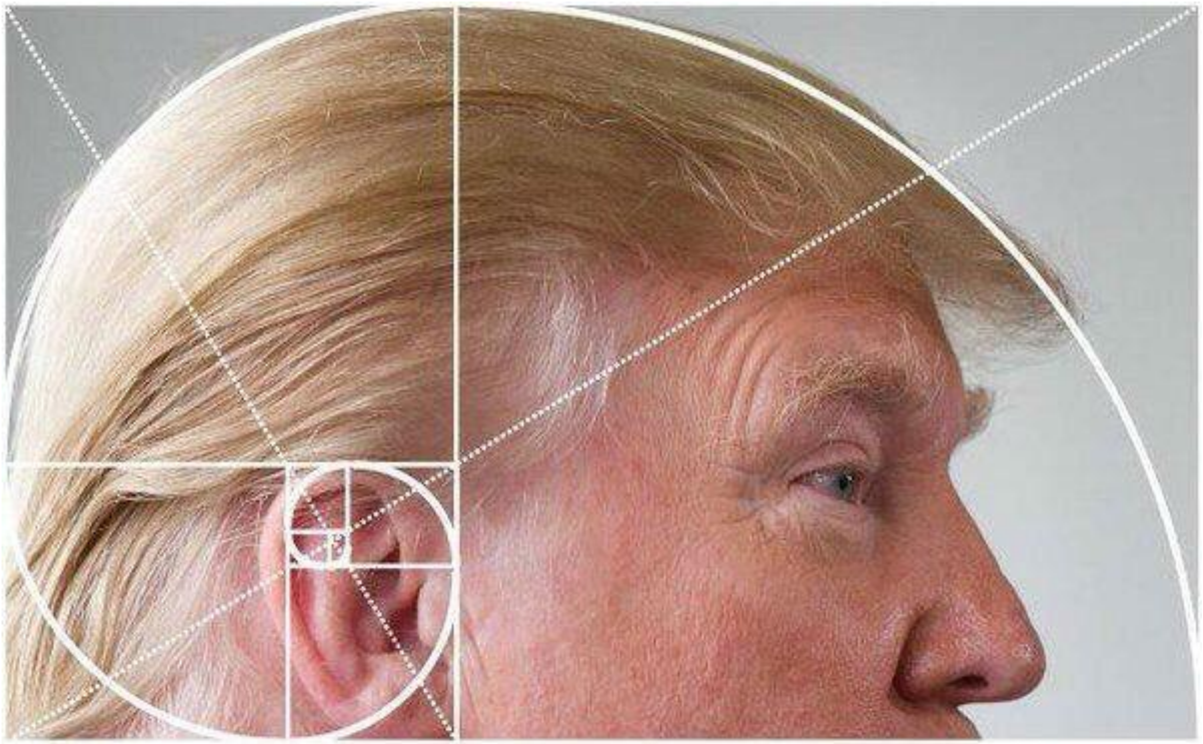


In mathematical terms, the sequence F_n of Fibonacci numbers is defined by the recurrence relation

$$F_n = F_{n-1} + F_{n-2},$$

with seed values

$$F_1 = 1,\ F_2 = 1$$



How much Fibonacci number fit into:

- Byte?

How much Fibonacci number fit into:

- Byte?

- Short?

How much Fibonacci number fit into:

- Byte?

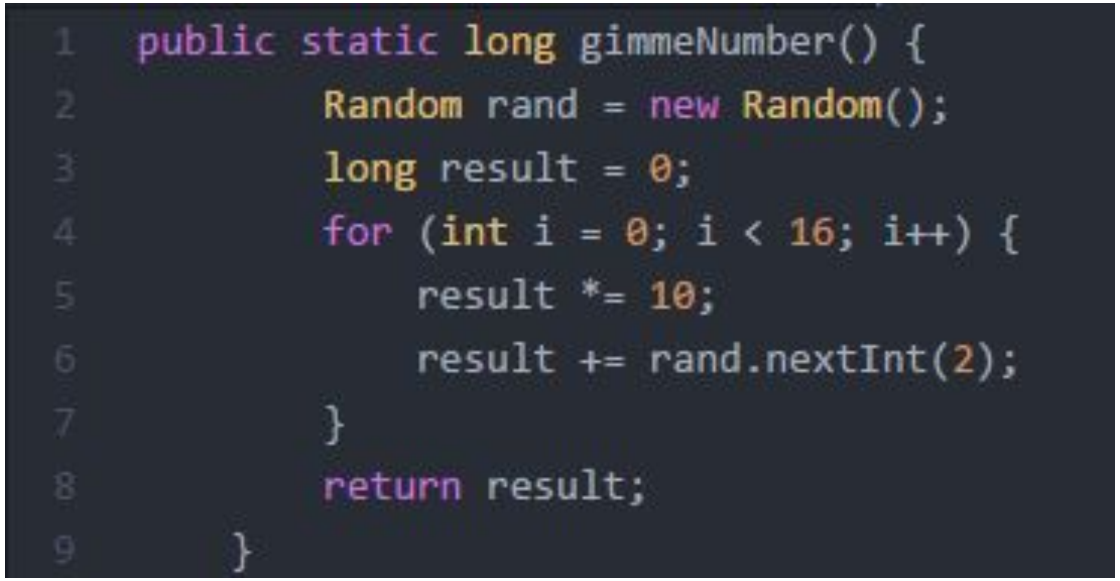
- Short?

- Int?

How much Fibonacci number fit into:

- Byte?
- Short?
- Int?
- Long?

Random numbers



```
public static long gimmeNumber() {
        Random rand = new Random();
        long result = 0;
        for (int i = 0; i < 16; i++) {
            result *= 10;
            result += rand.nextInt(2);
        }
        return result;
    }
```

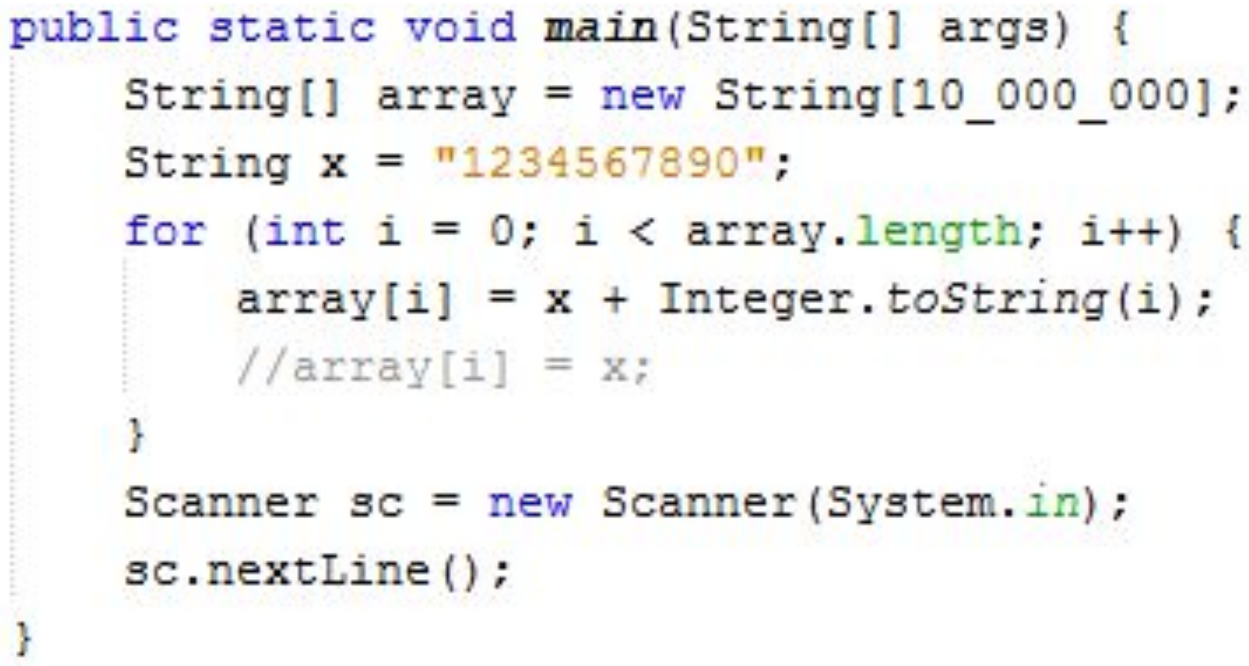
Unfortunately, this code is returning binary numbers as decimal integers, and you cannot fix the library itself, but you can write a fix, that takes the result of the function and converts it into regular integer. Use % and >> operations to complete the task.

Arrays

Initialize array with Pascal triangle. Print it to the screen.

```
           1
         1   1
       1   2   1
     1   3   3   1
   1   4   6   4   1
 1   5  10  10   5   1
```

String



```java
public static void main(String[] args) {
    String[] array = new String[10_000_000];
    String x = "1234567890";
    for (int i = 0; i < array.length; i++) {
        array[i] = x + Integer.toString(i);
        //array[i] = x;
    }
    Scanner sc = new Scanner(System.in);
    sc.nextLine();
}
```

Advanced

1. Multiply floating point numbers by 2 without floating point multiplication, but using bitwise operations and `Double.longBitsToDouble(long)`, `Double.doubleToRawLongBits(double)`

```
long ld = Double.doubleToLongBits(d);
    long sign = ld >> 63;
    long exp = (ld >> 52) & 0x7FF;
    long mantissa = ld & 0xFFFFFFFFFFFFFL;
    System.out.println(sign);
    System.out.println(exp - 1023);
    System.out.println(1.0 + mantissa);
```

Advanced

2. Numerical integration. Integrate f(x) = x^4 on the interval from -1000 to 0.
Use double.

Advanced

3. Write simple word counter. Tokenize, lower case and count.