

8.3 ガウス素数

複素数の登場で世界が変わってしまった。とくに変わってしまうのが素数の世界である。なぜなら、いままで2は素数の扱いをしていたが $2 = (1 + i)(1 - i)$ に分解できるからだ。しかし3はそうならない。3は複素数の範囲にまで広げても素数であり続ける。

実数の世界で素数であったものが、複素数の世界で合成数になってしまう数にはどんなものがあるだろうか。そのようになる実数 x は

$$x = (a + bi)(a - bi)$$

を満たしているはずである。つまり、共役複素数の積でないといけない。そういうものを見つけるには **Java** を頼ろうじゃないか。

programming list [GPcheck.java]

```

1: import java.util.Scanner;
2:
3: public class GPcheck {
4:
5:     public static void main(String[] args) {
6:         Scanner s = new Scanner(System.in);
7:
8:         System.out.print("input your candidate: ");
9:         int p = s.nextInt();
10:
11:         for(int i = 1; i <= Math.sqrt(p); i++) {
12:             double m = Math.sqrt(p - i * i);
13:             if (m == (int) m) {
14:                 System.out.println(p + " = (" +
15:                                     i + " + " + m + "i)(" + i + " - " + m + "i)");
16:             }
17:         }
18:         System.out.println("a Gauss prime.");
19:     }
20: }

```

プログラムは、素数 p が複素数の世界で分解できるか、できないのかを示すものである。しかし、このプログラムは危険な入力を避けられない。つまり、どんな p に対しても処理をしてしまうので、入力する p は確実に素数であることが分かってないと困る。それが心配なら、以前の `pchk`

関数を組み込むとよい。

ガウス素数を見つけるには、入力された p から順に $1*1$ 、 $2*2$ 、 $3*3$ 、 $4*4$ 、... を引いたとき、残りが平方数になっているかどうかで判定している。だから、調べる範囲は \sqrt{p} までで十分だ。もし、平方数を引いて平方数が残れば、それが求める積になるので出力すればよい。

`System.out.println`; 文がやかましいのは勘弁してもらいたい。

$x = (a + bi)(a - bi)$ に分解できるということは $x = (b + ai)(b - ai)$ にもできるということだから、ガウス素数でなければいくつかの分解の仕方がある。あまり自明の解を表示してもうるさいだけなので、`return`; 文で抜けている。全部の分解を表示したければ `return`; を削ってほしい。ところでこの `return`; は `for` ループを抜けるためでなく、`main()` 自体から抜けたいのだ。なぜなら、分解表示したらプログラムは目的を達成したのだから。そのため、`break`; には荷が重い処理なので使えなかったのだ。

`return`; が実行されなければ、それは `for` 構文の処理がひとしきり終わるまで続くことを意味する。`for` 構文の処理がひとしきり終わるといのは、調査範囲の最後まで計算してみたものの、素数が分解できなかった場合を指す。このときは一度も `return`; が実行されていないときで、ここではじめて “a Gauss prime.” と表示されるのである。

一般の複素数にも、分解できるものとできないものがある。分解できれば合成数で、できないものがガウス素数と呼ばれる。たとえば $4 + 3i = (1 + 2i)(2 - i)$ であるから合成数だ。一方、 $2 + 3i$ はガウス素数である。それはどこで分かるのだろうか。実は、 $a + bi$ に対して、 $a^2 + b^2$ が素数になればガウス素数であることが分かっている。実際、 $4 + 3i$ は $4^2 + 3^2 = 25$ (合成数)、 $2^2 + 3^2 = 13$ (素数) となっている。そういうことなら、ガウス素数の判定は簡単である。

programming list [GPcheck2.java]

```

1: import java.util.Scanner;
2:
3: public class GPcheck2 {
4:
5:     public static void main(String[] args) {
6:         Scanner s = new Scanner(System.in);
7:
8:         System.out.print("input 'a b' as a+bi: ");
9:         int re = s.nextInt();
10:        int im = s.nextInt();
11:

```

tmt's math page!

```
12:         int n = re * re + im * im;
13:
14:         String msg = "a Gauss prime.";
15:         for(int i = 2; i < n; i++) {
16:             if(n % i == 0) {
17:                 msg = "NOT a Gauss prime.";
18:             }
19:         }
20:         System.out.println(re + " + " + im + "i" + " is " + msg);
21:     }
22: }
```

複素数 $a + bi$ がガウス素数かどうかは、 $a^2 + b^2$ が素数かどうかにかかっているので、その判定はさほど効率的ではないけれど以前の PrimeChk プログラムがそのまま使える。だから main() メソッドは、入力された複素数 c の $a^2 + b^2$ を判定するだけのことである。

本当は複素数を $a+bi$ の形で入力したいけれど、簡便に済ますためなので、今回はこの仕様で勘弁してほしい。