

ملاحظات:

يمكننا في البرنامج السابق القيام بالأمر التالي:

\* يمكن طباعة حاصل جمع  $C1, C2$  مباشرة دون إرساء قيمة المجموع أولاً لـ  $C$  وذلك بالعلية:

```
C1.sum(C2).print();
```

وذلك لأن الدالة `sum` تعيد متحولاً من نوع `Complex`، ومن أجل أي كائن من النوع `Complex` يمكننا استدعاء دالة الطباعة `print()`.

\* يمكننا الاستغناء عن `m` والطرح باستخدام `C` نفسها كما يلي:

```
C = C1.minus(C2.R, C2.I); C.print();
```

ولكن هنا نكون قد أنشأنا كائناً جديداً في الذاكرة وفرضنا فيه قيمة الطرح ثم أسأنا عليه بـ `C`، بينما الاتزال قيمة `C` السابقة (قيمة المجموع) موجودة في الذاكرة، ولذا فقدنا أي وسيلة للوصول إليها.

\* طريقة ثانية لكاتبه الطباعة:

```
void print(Complex c) {
    if (c.I > 0)
        System.out.println(c.R + "+i" + c.I);
    else if (c.I < 0)
        System.out.println(c.R + "-i" + (-c.I));
    else System.out.println(c.R);
}
```

وبهذه الحالة يمكن طباعة مجموع العددين، مثلاً، في البرنامج كما يلي:

```
C1.print(C1.sum(C2));
```

وهنا يمكن استبدال `C1` الأول بأي متحول آخر من نوع الصنف `Complex` وسنحصل على نفس النتيجة، ولكن وجوده ضروري لأنه لا يمكن استدعاء

الدالة `print` (أدأى من الدوال الأخرى في الصنف) إلا عن طريق متحول من نوع `Complex`.