

**1)** Die *imaginäre* Einheit  $i$  ist die Wurzel aus  $-1$  oder genauer ausgedrückt:  $i^2 = -1$ . Imaginäre (oder auch: komplexe) Zahlen haben einen **Realteil** und einen **Imaginärteil**:  $z = 3 + 4i$  ist eine Zahl, oder allgemein

$y = a + b \times i$  (wobei  $a$  und  $b$  reelle Zahlen sind).

**Die Rechenregeln für imaginäre Zahlen sind einfach die für geklammerte Ausdrücke!**

Für das Folgende legen wir fest:  $y = a+bi$ ,  $z = c+di$ , wobei  $a$ ,  $b$ ,  $c$  und  $d$  reelle Zahlen und  $i$  die imaginäre Einheit sind.

**Addition  $y+z$ :**  $(a + bi) + (c + di) = (a+c) + (b+d)i$

**Subtraktion  $y-z$ :**  $(a + bi) - (c + di) = (a-c) + (b-d)i$

**Multiplikation:**

mit einem Skalar (reelle Zahl)  $k$ :  $k \times y = k \times (a + bi) = ka + kbi$

mit einer komplexen Zahl:  $y \times z = (a + bi) \times (c + di) = ac + adi + bci + bdi^2$  und weil  $i^2 = -1$  weiter:  $(ac - bd) + (ad + bc)i$

**Division:**

Durch ein Skalar  $k$ :  $y \div k = (a \div k) + (b \div k)i$

$$\frac{y}{z} = \frac{a+bi}{c+di} = \frac{a+bi}{c+di} \times 1 = \frac{a+bi}{c+di} \times \frac{c-di}{c-di} = \frac{(a+bi) \times (c-di)}{(c+di) \times (c-di)} = \frac{ac - adi + bci - bdi^2}{c^2 - cdi + cdi - d^2i^2}$$

$$= \frac{ac + bd + (bc - ad)i}{c^2 + d^2} = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2}i$$

Der **Absolutwert** ist der Abstand vom Nullpunkt:  $|y| = \sqrt{a^2 + b^2}$

*Hinweis: Die Rechenwege sind jeweils nur gezeigt, um sich überzeugen zu können – zum Programmieren brauchen Sie jeweils immer nur das fett gedruckte Ergebnis hinter dem letzten „=“!*

Erstellen Sie nebenstehende Klasse für imaginäre Zahlen (Unterstreichung bedeutet `>static<`, `+` steht für `>public<`, `-` für `>private<`)

- Die Methode `toString()` soll den String " $a + bi$ " retournieren (wobei  $a$  und  $b$  natürlich durch ihre Werte ersetzt sind).
- Die Methode `main()` soll wenigstens zwei imaginäre Zahlen erzeugen und Testausgaben für alle Methoden ausgeben.

(3 Punkte)

ComplexNumber
-a : float
-b : float
+getRealPart() : float
+getImaginaryPart() : float
+toString() : String
+abs() : float
+add( ComplexNumber ) : ComplexNumber
+subtract( ComplexNumber ) : ComplexNumber
+multiplyBy( float ) : ComplexNumber
+multiplyBy( ComplexNumber ) : ComplexNumber
+divideBy( float ) : ComplexNumber
+divideBy( ComplexNumber ) : ComplexNumber
+main( String[] ) : void

**2)** In Aufgabe 1 haben Sie eine Klasse erstellt, deren Objekte komplexe Zahlen darstellen.

Entwerfen und programmieren Sie dazu eine **Bildschirmmaske**, in die man die Werte für 2 solcher Objekte eingeben kann. In der Gestaltung sind Sie völlig frei, aber volle Punkte (100%) erhalten Sie für ein JFrame-Form mit GridBag-Layout. Für Null-Layout oder FreeDesign (oder eines der anderen „einfachen“ Layouts, die allerdings für die Aufgabe ohnehin unbrauchbar sein dürften), gibt es bis zu 60%.

Einen **Zusatzpunkt** können Sie sich holen, indem Sie Ihr Form skalierbar machen!

Ihre Maske soll auch Buttons enthalten, die die Operationen Ihrer ›Arbeitsklasse‹ aufrufen. Bei Klick erzeugen Sie die jeweils benötigte Anzahl an Objekten mit entsprechenden Werten, rufen die jeweiligen Methoden auf und zeigen das Ergebnis auf der Maske an.

(6 Punkte)