

Πρώτη Σειρά ασκήσεων Ημερομηνία Παράδοσης: 21 Απριλίου 2019, 21:00 μ.μ.

Στην άσκηση αυτή θα σχεδιάσετε και θα υλοποιήσετε σε JAVA ένα πρόγραμμα που θα περιγράφει ένα μιγαδικό αριθμό και θα εκτελεί πράξεις μεταξύ μιγαδικών αριθμών.

Η άσκηση έχει δύο κομμάτια. Στο πρώτο κομμάτι θα υλοποιήσετε μία κλάση `ComplexNumber` που θα περιγράφει ένα μιγαδικό αριθμό, του οποίου το πραγματικό όσο και το φανταστικό μέρος θα αποτελείται από αριθμούς κινητής υποδιαστολής τύπου `double`. Στο δεύτερο κομμάτι θα υλοποιήσετε μία κλάση `ComplexMatrix` η οποία θα περιγράφει ένα τετραγωνικό ($N \times N$) δισδιάστατο πίνακα από μιγαδικούς αριθμούς και θα υλοποιεί τις πράξεις της πρόσθεσης και της αφαίρεσης οποιουδήποτε τετραγωνικού πίνακα.

Υπενθύμιση:

- Για την υλοποίηση των πράξεων της πρόσθεσης και της αφαίρεσης οι πίνακες που εμπλέκονται θα πρέπει να έχουν **τον ίδιο αριθμό γραμμών και στηλών**.
- Στα μαθηματικά, οι μιγαδικοί αριθμοί είναι μία επέκταση του συνόλου των πραγματικών αριθμών με την προσθήκη του στοιχείου i , που λέγεται φανταστική μονάδα, και έχει την ιδιότητα: $i^2 = -1$
- Κάθε μιγαδικός αριθμός μπορεί να γραφτεί με τη μορφή $a + ib$, όπου τα a και b είναι πραγματικοί αριθμοί και λέγονται πραγματικό μέρος και φανταστικό μέρος του μιγαδικού αριθμού, αντίστοιχα. Για παράδειγμα, ο $3 + 2i$ είναι ένας μιγαδικός, με πραγματικό μέρος 3 και φανταστικό μέρος 2.
- Για περισσότερες πληροφορίες σχετικά με τους μιγαδικούς αριθμούς μπορείτε να διαβάσετε το σχετικό υλικό από το wikipedia.

Το δεύτερο κομμάτι είναι υπερσύνολο του προηγούμενου: θα πρέπει να προσθέσετε επιπλέον πεδία και μεθόδους στις κλάσεις σας, και να υλοποιήσετε επιπλέον κλάσεις. Θα βαθμολογηθείτε μόνο για τα κομμάτια που έχετε υλοποιήσει αν δεν έχετε καταφέρει να ολοκληρώσετε επιτυχώς όλα τα κομμάτια.

Το πρώτο κομμάτι το κάνετε **turnin στο `assignment1.1@myy205`** και το δεύτερο στο **`assignment1.2@myy205`**

Πρώτο μέρος

Σε ένα αρχείο `ComplexNumber.java` θα υλοποιήσετε μία κλάση `ComplexNumber` που θα περιγράφει ένα μιγαδικό αριθμό. Η κλάση περιέχει δύο `private` πεδία τύπου `double`, ένα που αντιπροσωπεύει το πραγματικό μέρος του αριθμού και θα ονομάζεται `real` και ένα που αντιπροσωπεύει το φανταστικό μέρος και θα ονομάζεται `img`. Επιπλέον, η κλάση υλοποιεί τον εξής constructor:

- `ComplexNumber(double real, double img)`: το πρώτο όρισμα αντιπροσωπεύει το πραγματικό μέρος του αριθμού και το δεύτερο όρισμα το μιγαδικό μέρος.

Η κλάση θα διαθέτει τις κατάλληλες `public` μεθόδους για την πρόσβαση (**accessor**) και την ανάθεση (**mutator**) όλων των πεδίων της κλάσης. Επιπλέον, διαθέτει τις παρακάτω μεθόδους:

1. `ComplexNumber addComp(ComplexNumber num)`: Θα παίρνει ως όρισμα ένα αντικείμενο μιγαδικό αριθμό `num` και θα τον προσθέτει στο μιγαδικό αριθμό που θα καλεί τη μέθοδο αυτή.
2. `ComplexNumber subtractComp(ComplexNumber num)`: Θα παίρνει ως όρισμα ένα αντικείμενο μιγαδικό αριθμό `num` και θα τον αφαιρεί από το μιγαδικό αριθμό που θα καλεί τη μέθοδο αυτή.
3. `ComplexNumber multiplyComp(ComplexNumber num)`: Θα παίρνει ως όρισμα ένα αντικείμενο μιγαδικό αριθμό `num` και θα τον πολλαπλασιάζει με το μιγαδικό αριθμό που θα καλεί τη μέθοδο αυτή.
4. `String toString()`: Επιστρέφει τα περιεχόμενα του μιγαδικού αριθμού σε ένα αντικείμενο τύπου `String` ως εξής:

Real Symbol Image i, όπου **Real** το προσημασμένο πραγματικό μέρος του μιγαδικού αριθμού, **Symbol** το πρόσημο του φανταστικού μέρους του αριθμού και **Image** η απόλυτη τιμή του φανταστικού μέρους του αριθμού και **i** σταθερά που εμφανίζεται σε όλες τις εκτυπώσεις. Κάποια παραδείγματα φαίνονται παρακάτω:

```
3.51 + 1.87i
2.35 - 5.61i
-8.45 + 2.65i
```

5. `boolean equals(ComplexNumber num)`: Θα παίρνει ως όρισμα ένα αντικείμενο μιγαδικό αριθμό `num` και θα ελέγχει την ισότητα δύο μιγαδικών αριθμών και θα επιστρέφει μία `boolean` τιμή.

Δεύτερο μέρος

Σε ένα αρχείο `ComplexMatrix.java` θα υλοποιήσετε μία κλάση `ComplexMatrix` που θα αποτελείται από ένα πίνακα μεγέθους `MxN` τύπου `ComplexNumber` οποίος θα πρέπει να δηλωθεί ως `private` πεδίο της κλάσης. Η κλάση διαθέτει τους εξής `constructor`:

- `ComplexMatrix()`: Δημιουργεί ένα `default constructor`.
- `ComplexMatrix(ComplexMatrix original)`: Δημιουργεί ένα αντικείμενο τύπου `ComplexMatrix` που αποτελεί αντίγραφο του πίνακα `original` που δίνεται ως όρισμα. Τα στοιχεία του νέου πίνακα, είναι αντίγραφα των στοιχείων του πίνακα `original`.
- `ComplexMatrix(int rows, int cols)`: Δημιουργεί ένα τυχαίο δισδιάστατο πίνακα μεγέθους `rows x cols`, από αντικείμενα τύπου `ComplexNumber`. Για την παραγωγή τυχαίων αριθμών μπορείτε να χρησιμοποιήσετε την κλάση `Random` κάνοντας `import` τη βιβλιοθήκη `java.util.Random`. Για την παραγωγή ενός τυχαίου αριθμού στο διάστημα `-10` έως `10` μπορείτε να χρησιμοποιήσετε τον παρακάτω κώδικα που σας δίνεται:

```
import java.util.Random;

class ComplexMatrix
{
    private Random rand = new Random();

    private double computeRandom()
    {
        int randomNum = (int)((rand.nextDouble()-0.5)*rand.nextInt(20)*100);
        return randomNum/100.0;
    }
}
```

Η κλάση `ComplexMatrix` θα διαθέτει επίσης τις παρακάτω `public` μεθόδους:

- `String toString()`: Επιστρέφει τα περιεχόμενα του πίνακα σε ένα `String`. Θα πρέπει να χρησιμοποιεί τη μέθοδο `toString()` της κλάσης `ComplexNumber` για την εκτύπωση των μιγαδικών αριθμών. Στην αρχή του πίνακα εκτυπώνεται ο χαρακτήρας "[". Ανάμεσα σε δύο μιγαδικούς αριθμούς που ανήκουν στην ίδια γραμμή παρεμβάλλεται χαρακτήρας κόμμα "," ακολουθούμενος από τον κενό χαρακτήρα. Στο τέλος κάθε γραμμής θα εκτυπώνεται ο χαρακτήρας ";" ακολουθούμενος από χαρακτήρα αλλαγής γραμμής. Στο τέλος του πίνακα εκτυπώνεται ο χαρακτήρας "]" ακολουθούμενος από χαρακτήρα αλλαγής γραμμής. Ένα παράδειγμα εκτύπωσης ενός πίνακα διαστάσεων 3x3 φαίνεται παρακάτω:

```
[1.24 + 2.55i, -0.32 + 2.00i, 1.35 - 5.88i;
-5.71 - 5.91i, 0.29 - 9.14i, 0.00 + 3.51i;
6.44 + 0.00i, -3.51 - 0.67i, 2.10 + 4.20i;]
```

- `ComplexMatrix CompAdd(ComplexMatrix matrix)`: Επιστρέφει ένα νέο πίνακα που αποτελείται από το άθροισμα του πίνακα του αντικειμένου που την κάλεσε με τον πίνακα `matrix`. Εάν οι δύο πίνακες δεν έχουν τις ίδιες διαστάσεις πρέπει να επιστρέφει `null`.
- `ComplexMatrix CompSubtract(ComplexMatrix matrix)`: Επιστρέφει ένα νέο πίνακα που αποτελείται από τη διαφορά του πίνακα του αντικειμένου που την κάλεσε από τον πίνακα `matrix`. Εάν οι δύο πίνακες δεν έχουν τις ίδιες διαστάσεις πρέπει να επιστρέφει `null`.

Οδηγίες υλοποίησης

Στην υλοποίηση των κλάσεων σας δεν θα πρέπει να έχετε `public` πεδία. Βαθμοί θα αφαιρεθούν για προγράμματα που δεν είναι καλά γραμμένα, δηλαδή δεν είναι σωστά στοιχισμένα ή δεν έχουν καλά επιλεγμένα ονόματα μεταβλητών ώστε να διαβάζονται εύκολα.

Προγράμματα που δεν κάνουν `compile` θα πάρουν το πολύ 1 μονάδα από τις 10. Θα πάρετε περισσότερες μονάδες αν έχετε ολοκληρωμένες σωστές μεθόδους, ή μια ολόκληρη κλάση η οποία είναι σωστή, παρά αν έχετε πολύ κώδικα που όμως δεν κάνει `compile` ή δεν τρέχει.

Για τον έλεγχο των κλάσεων που έχετε δημιουργήσει μπορείτε να χρησιμοποιήσετε τον κώδικα που σας δίνεται.

Τα αρχεία που θα παραδώσετε θα ελεγχθούν για αντιγραφές μέσω ειδικού λογισμικού. Οι αντιγραφές μηδενίζονται για όλα τα εμπλεκόμενα άτομα.

Στον κώδικα να αναγράφονται σε σχόλια το όνομα, το login και ο ΑΜ σας. Χρησιμοποιείτε μόνο λατινικούς χαρακτήρες αλλιώς δεν λειτουργεί το turnin. «Ορφανές» ασκήσεις δεν θα βαθμολογούνται.

Παράδειγμα αποστολής ασκήσεων:

1^ο μέρος: turnin assignment1.1@myy205 ComplexNumber.java

2^ο μέρος: turnin assignment1.2@myy205 ComplexMatrix.java