

## Metode Avansate de Programare, 2022–2023, LABORATOR 1

### Obiective:

- instalarea mediilor de dezvoltare necesare pentru a scrie programare folosind limbajul Java
- familiarizarea cu sintaxa limbajului Java

**Obs. Acest laborator nu intra în calculul notei de la laborator, dar sunteți încurajați să realizați cerințele de mai jos.**

A. Instalați un mediu de dezvoltare integrat pt Java (Java IDE ) pe laptopurile proprii și încercați să vă familiarizați cu acesta.

*La curs și seminar vom folosi:* IntelliJ IDEA, Ultimate Edition (Pentru a putea activa licența, creați un JetBrains Account cu adresa de mail de la facultate).

[Download JDK]

<https://www.oracle.com/java/technologies/downloads/>

[Download IntelliJ IDEA]

<https://account.jetbrains.com/login>

B. Se citește, ca parametri în linia de comandă, separați prin spațiu, mai multe numere complexe de forma  $a+bi$  și o operație, sub forma unui operator (+, -, \*, /).

1. Să se verifice dacă parametri citați în linia de comandă, separați prin spațiu, reprezintă o expresie aritmetică de forma:

$$n_1 \text{ op } n_2 \text{ op } \dots \text{ op } n_k,$$

unde  $n_1, n_2 \dots n_k$  sunt numere complexe de forma  $a + b * i$ , iar  $op$  este operatorul dat.

Exemplu:  $2 + 3 * i + 5 - 6 * i + -2 + i$

$\text{args}[0] = 2 + 3 * i$ ,  $\text{args}[1] = +$ ,  $\text{args}[2] = 5 - 6 * i$ ,  $\text{args}[3] = +$ ,  $\text{args}[4] = -2 + i$

2. Dacă parametri citați în linia de comandă reprezintă o expresie aritmetică de forma descrisă la punctul 1, se cere să se afișeze rezultatul acestei expresii.

Exemplu: pentru expresia dată se va afișa:  $5 - 2 * i$

Observatii:

- 1) Sunteți încurajați să veniți cu propria proiectare pentru rezolvarea problemei dacă aceasta este corectă și respectă principiile de proiectare orientată obiect învățate până acum (Vezi SOLID design principles). **O proiectare simplistă (totul într-un fișier, fără clase specializate ....) care doar rezolvă problema va fi punctată cu nota 8.**

- 2) Sugestii de proiectare (daca nu va este teama sa va jucati cu clasele, puteti veni si cu propria abstractizare, chiar sunteti incurajati).

Se vor defini clasele:

- *NumarComplex*, având ca attribute *re*, *im* de tip real, iar ca metode operatiile definite pe multimea numerelor complexe: *adunare*, *scadere*, *inmultire*, *impartire* si *conjugatul*.
- *ComplexExpression*, clasa **abstracta**, ce are ca attribute *operation* de tipul *Operation* (enum având patru valori posibile: **ADDITION**, **SUBTRACTION**, **MULTIPLICATION**, **DIVISION**) si un vector (array) *args* de numere complexe, iar ca metode, o metodă *execute* ce returneaza rezultatul expresiei aritmetice, apeland pentru aceasta metoda **abstractă** *executeOneOperation* [TemplateMethodDesignPattern]. Derivati din clasa *ComplexExpression* clasele necesare pentru executia celor patru tipuri de expresii, definite la punctul B.1., suprascriind, corespunzator tipului operatiei dat (+,-,/,\*), metoda *executeOneOperation* - executa o operatie din insiruirea de operatii ale expresiei.
- *ExpressionFactory* (singleton) [SingletonDesignPattern] [FactoryMethodDesignPattern] având metoda *createExpression* cu semnatura: **public** ComplexExpression createExpression(Operation operation, Complex[] args) si care creează o expresie în funcție de valoarea parametrului *operation* (vezi diagrama de clase).
- *ExpressionParser* care parsează expresia, verifică dacă este validă si construiește, folosind un obiect de tipul ExpressionFactory, expresia corespunzatoare operatorului dat.

### Alte Referinte:

A se vedea si cursul si seminarul 1.

[FactoryMethodDesignPattern] [https://www.tutorialspoint.com/design\\_pattern/factory\\_pattern.htm](https://www.tutorialspoint.com/design_pattern/factory_pattern.htm)

[SingletonDesignPattern] [https://www.tutorialspoint.com/design\\_pattern/singleton\\_pattern.htm](https://www.tutorialspoint.com/design_pattern/singleton_pattern.htm)

[TemplateMethodDesignPattern] [https://www.tutorialspoint.com/design\\_pattern/template\\_pattern.htm](https://www.tutorialspoint.com/design_pattern/template_pattern.htm)