

Documentatie – Tema 1

Calculator - Polinoame

Nume: Dascal Raluca Georgiana

Grupa: 30226

Profesor coordonator: Dorin Moldovan

Cuprins:

| | |
|---|---|
| 1. Cerinte functionale | 3 |
| 2. Obiective | 3 |
| 2.1 Obiectiv Principal | 3 |
| 2.2 Obiective Secundare | 3 |
| 3. Descrierea Interfetei Grafice | 3 |
| 4. Analiza Problemei | 4 |
| 5. Proiectare | 5 |
| 5.1 Diagrame de clase | 5 |
| 5.2 Descrierea Algoritmilor Implementati | 5 |
| 6. Implementare | 6 |
| 7. Rezultate | 8 |
| 8. Concluzii si Posibilitati de Dezvoltare Ulterioare | 9 |
| 9. Bibliografie..... | 9 |

1. Cerinte Functionale

Sa se dezvolte un calculator de polinoame (realizand si o interfeta grafica) care este capabil sa efectueze urmatoarele operatii matematice: adunare, scadere, inmultire, impartire, derivare si integrare.

2. Obiective

2.1 Obiectiv Principal

Dezvoltarea unei interfete grafice care sa permita citirea dintr-un camp de text a unui, respectiv a doua polinoame cu o singura necunoscuta si coeficienti de tip intreg, introduse de catre un utilizator si afisarea rezultatului corect din punct de vedere matematic in campul de text indicat in fereastra grafica in functie de operatia selectata.

Se cere si respectarea unor restrictii la nivel de cod (crearea unor clase cu maxim 300 de linii, metode cu un numar maxim de 30 de linii de cod, inlocuirea vectorilor cu liste, folosirea paradigmelor programarii orientate pe obiect (precum incapsulare) si inlocuirea instructiunii for cu instructiunea specifica listelor foreach).

2.2 Obiective Secundare

| Obiective secundare | Capitol |
|-----------------------------------|---------|
| Descrierea interfetei grafice | 3 |
| Descrierea algoritmilor utilizati | 5.2 |
| Prezentarea claselor | 6 |
| Rezultatele obtinute | 7 |

3. Descrierea Interfetei Grafice

Calculator Polinoame

Date de intrare trebuie introduse dupa formatul urmator: $a_1x^n+a_2x^{n-1}+...+a_{n-1}x^1+a_nx^0$

Introduceti polinoamele

Rezultatul e

Fig 3.1

In figura 3.1 este prezentata interfata grafica a aplicatiei create. Am ales sa dezvolt o astfel de interfata, adica una prietenoasa, cu scopul de a veni in ajutorul utilizatorului, pentru ca folosirea calculatorului sa fie cat mai facila pentru el. Am inclus in fereastra grafica doua campuri de text in care utilizatorul trebuie sa introduca cele doua polinoame, a caror rezultat este interesat sa-l determine. Atentie! Polinoamele ce sunt incluse in aceste campuri trebuie sa respecte urmatoarele conditii pentru a fi corect “receptionate” de catre program pentru a face totodata posibila si prelucrarea acestora: polinoamele trebuie sa fie introduse sub forma unor sume de monoame a caror forma sa corespunda urmatorului tipar: coeficient (acesta poate sa fie omis in cazul in care este ± 1), variabila necunoscuta (aceasta poate fi reprezentata prin orice litera mica a alfabetului englez), caracterul ^ (corespunzator ridicarii la un exponent) si putere (aceasta trebuie specificata inclusiv pentru valorile 0 si 1 ale puterilor) scrise fara spatii. Un exemplu de polinom introdus corect ar putea fi “ $2x^3-3x^1+13x^0$ ”. Dupa introducerea celor doua polinoame utilizatorul trebuie sa acceseze unul dintre butoanele incadrate in interfata grafica (etichetate cu simboluri corespunzatoare unor operatii matematice) in functie de operatie pe care acesta vrea sa o efectueze pe cele doua polinoame. Rezultatul obtinut in urma parcurgerii algoritmului implementat asociat etichetei butonului ales va fi afisat in campul de text indicat dupa eticheta “Rezultatul e”. Daca utilizatorul va apasa butonul cu eticheta “/” in campul de text va fi afisat un mesaj de eroare “Operatie necunoscuta” deoarece in acest proiect nu a fost implementat un algoritm care sa fie capabil sa returneze un rezultat corect pentru impartirea a doua polinoame, avand in vedere complexitatea acestei operatii matematice.

4. Analiza problemei

In figura de mai jos (4.1) este descrisa intr-o forma grafica relatia dintre utilizator,,interfata si metodele implementate cu ajutorul unor simboluri sugestive pentru a fi mai usor de inteles pentru un utilizator oarecare .

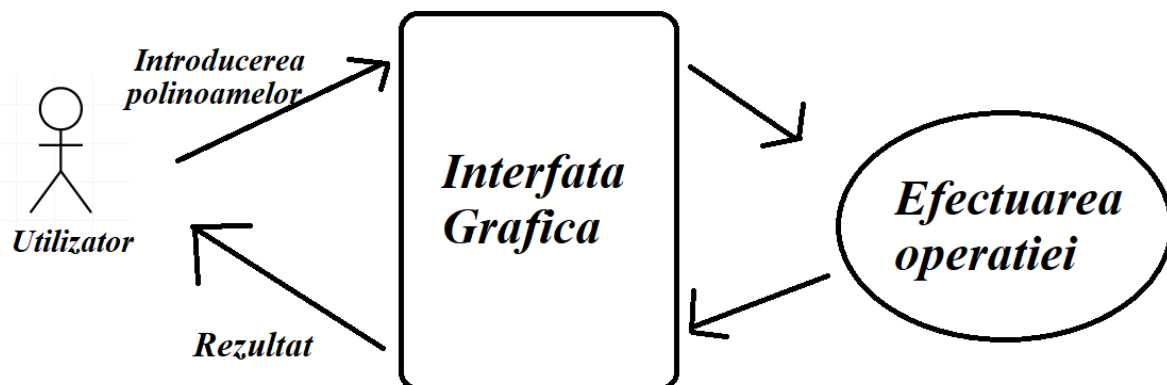


Fig 4.1

5. Proiectare

5.1 Diagrama de clase

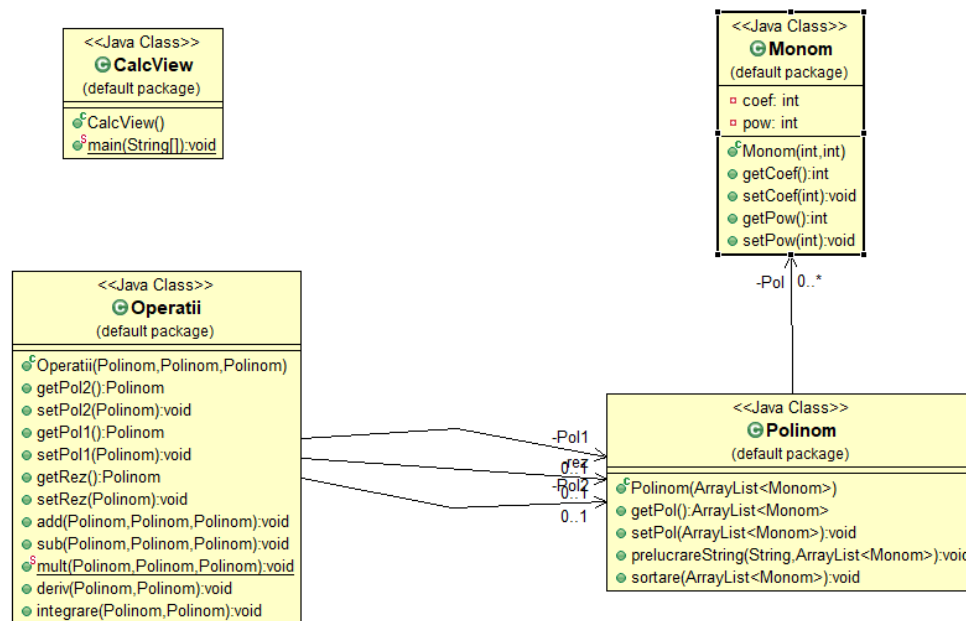


Fig 5.1.1

În figura 5.1.1 sunt prezentate diagramele UML a claselor proiectului realizat. Relația dintre acestea a fost sintetizată de către programul în care a fost scris și codul, Eclipse Java 2019. În această imagine se evidențiază atât fiecare clasă în parte (cu variabilele declarate și metodele ce definesc fie constructorii, gettere, settere, fie algoritmi utili în realizarea funcționalităților aparatului de calcul) cât și legăturile dintre acestea. Aceste diagrame UML ne ajută la formarea unei imagini în ansamblu a proiectului ajutând totodată și la înțelegerea unor concepte și paradigme a programării orientate pe obiect.

5.2 Descrierea algoritmilor implementați

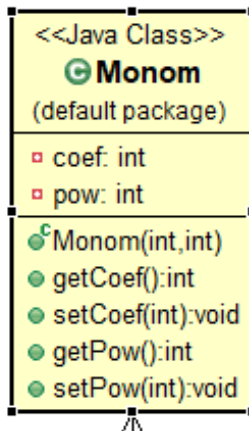
Fiecare metodă definită și descrisă în acest proiect, cu excepția constructorilor sau a celor care setează și returnează valorile variabilelor declarate în clase, descriu câte o sarcină pe care proiectul trebuie să o îndeplinească folosind un algoritm mai mult sau mai puțin riguros. Metoda `prelucrareString` conține un algoritm care permite “spargerea” unei variabile `String` și crearea unor polinoame prin adăugarea într-o listă a monomelor determinate folosind o tehnică “muncitorească” și nu una pe baza unor funcții predefinite în acest limbaj de programare (precum funcția `Split()`). Metoda `sortare` din clasa `Polinoame` este descrisă cu ajutorul unui algoritm simplu atât din punct de vedere sintactic cât și la nivel de programare ce are la bază funcția `Collection Sort` predefinită în pachetul `java.util.arraylist`. Metodele descrise în clasa `Operatii` sunt definiții ale unor operații matematice (adunare, scădere, înmulțire, împărțire, derivare și integrare) pe polinoame și conțin de asemenea algoritmi simpli din punct de vedere rațional ce au la bază strict concepte și reguli aparținând domeniului matematic. Metoda “sub” corespunzătoare scăderii este dezvoltată cu ajutorul metodei `add` (asociată adunării) - acest lucru se realizează prin înmulțirea fiecărui

coeficient al monoamelor din cel de al doilea polinom cu -1 dupa care se aplica algoritmul implementat in metoda add pe primul polinom introdus si cel nou creat (prin inversare signaturii coeficientului)).

6. Implementare

In acest capitol se va prezenta fiecare clasa a proiectului si se va explica fiecare variabila declarata si metoda implementata in acestea avand la baza diagrama UML sintetizata de catre programul in care a fost scris codul. S-a pastrat un unic pachet care contine toate cele patru clase create (default package).

6.1. Clasa Monom



In figura alaturata (6.1) este prezentata clasa Monom care reprezinta clasa de temelie a proiectului. Aceasta sta la baza intregi lucrari continand declararea unui monom (definit prin coeficient si putere, variabile de tip intreg). Aceasta clasa a fost construita cu scopul de a ajuta la crearea polinoamelor (obiecte elementare in realizarea acestui proiect) ca liste de monoame. In aceasta clasa s-a definit un constructor explicit si s-au implementat metodele elementare in acest limbaj de programare, adica gettere si settere pentru fiecare variabila declarata.

Fig 6.1

6.2. Clasa Polinom

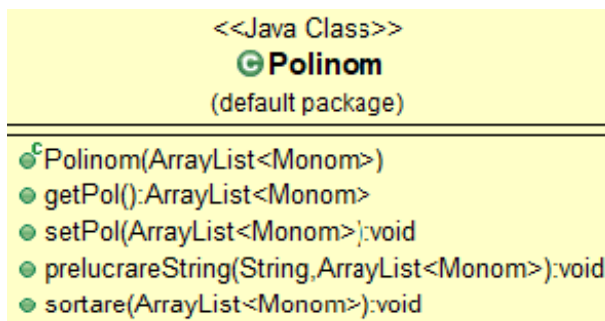


Fig 6.2
fiecaruia dintre acestea

Clasa Polinom, descrisa in figura 6.2 contine o singura variabila, un polinom care a fost definit ca o lista de monoame. Am creat si aici un constructor explicit a clasei. Pe langa uzualele gettere si settere in aceasta clasa am descris si urmatoarele doua metode:

- prelucrareString - metoda care este folosita la impartirea polinomului primit ca un text in monoame (cu coeficient si putere) si adaugarea acestora intr-o lista declarata astfel incat sa fie posibila retinerea celor doi paramterii cu scopul formarii polinomului,

- sortare - aceasta metoda presupune ordonarea monoamelor fiecarui polinom introdus in ordine descrescatoare tinand cont de puterea

Metoda de sortare este necesara si totodata utila deoarece nu mai este obligatoriu sa se impuna vreo regula in introducerea polinomului de catre utilizator (nu se cere scrierea monoamelor in ordine descrescatoare a gradelor) astfel facand cat mai usoara utilizarea calculatorului propus.

6.3. Clasa Operatii

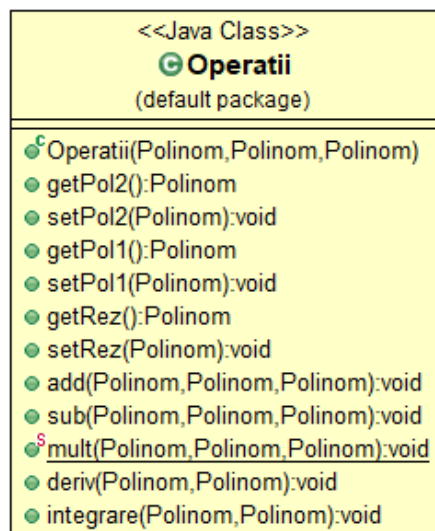
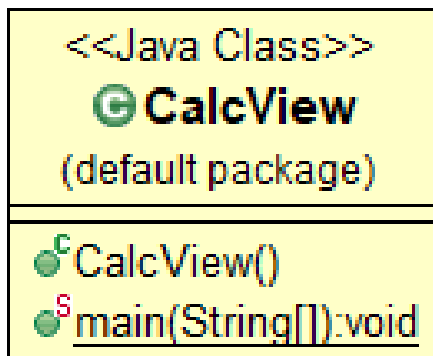


Fig 6.3

parametrii: unul dintre ei a fost prezentat si mai sus (polinomul asupra caruia se efectueaza operatia matematica), iar celalalt reprezinta si aici rezultatul obtinut in urma realizarii operatiei matematice. ! Afisarea rezultatului operatiei de integrare a fost realizata printr-un "artificiu" la nivelul formarii string-ului ce va fi returnat in campul de text folosind fractii (adica caracterul "/") pentru a mentine tipul atributului coeficient (int) al monoamelor declarant in clasa cu acelasi nume.

In clasa Operatii (prezentata in figura 6.3) sunt declarate doar 3 variabile, toate de tip polinom - 2 dintre ele fiind reprezentate de polinoamele introduse de catre utilizator in campurile de text indicate in interfata grafica, iar cel de al treilea reprezinta rezultatul obtinut in urma parcurgerii algoritmului sugerat de catre eticheta butonului selectat. Si in aceasta clasa sunt prezente atat gettere si settere cat si constructorul explicit. Pe langa acestea este descrisa fiecare operatie matematica printr-o metoda diferita. Metodele add, sub, mult corespundatoare operatiilor adunarii, scaderii respectiv inmultirii (in aceasta ordine) primesc ca parametrii cele 2 polinoame introduse in campurile text de catre utilizator si inca unul in care se va retine rezultatul obtinut in urma efectuarii operatiei alese care va fi si afisat in fereastra grafica in campul de text indicat. Spre deosebire de aceste metode, celalalte doua precum deriv si integrare corespundatoare derivarii respectiv integrarii necesita un singur polinom asupra caruia se va executa algoritmul descris fiecarei metode reusind si returnarea unei valori drept rezultat. Aceste metode contin doar doi

6.4. Clasa CalcView



in Fig 6.4

Clasa CalcView prezentata in figura alaturata (Fig 6.4) este clasa principala a proiectului, cea care contine si procedura de baza Main(). In aceasta clasa este realizata si crearea interfetei grafice (declararea tuturor componentelor din interfata grafica - campurile de text, butoanele, etichetele precum si definirea unei aranjari a acestora intr-o rama) si sunt definiti ascultatori pentru fiecare buton apeland metodele din clasa Operatii specifice caracterului de pe eticheta fiecarui buton. Clasa CalcView este clasa care intruneste si foloseste toate celalalte clase si toate metodele definite precedent in acest proiect. Datorita acestui fapt este posibila realizarea functionalitatii acestui proiect - calculator de polinoame - adica este executata corect atat citirea celor doua polinoame introduse de catre utilizator in campurile de text, cat si furnizarea unui rezultat obtinut prin efectuarea unor calcule corecte din punct de vedere matematic in functie de cerintele utilizatorului (a butonului accesat).

7. Rezultate

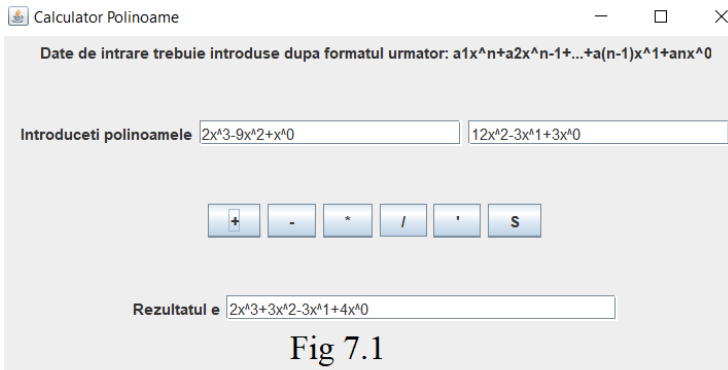


Fig 7.1

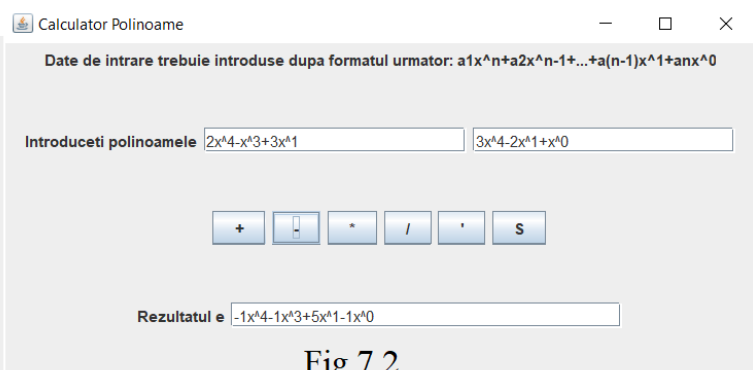


Fig 7.2

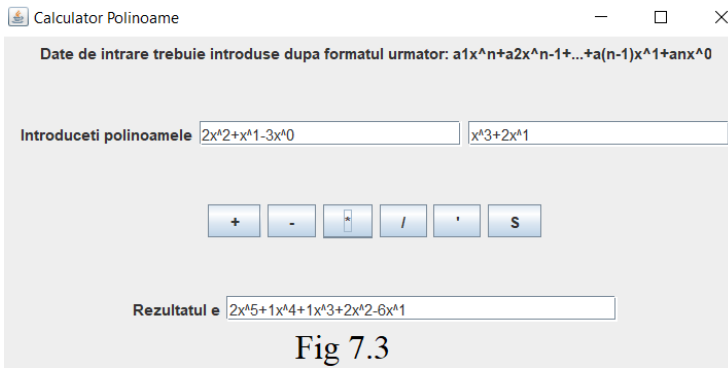


Fig 7.3

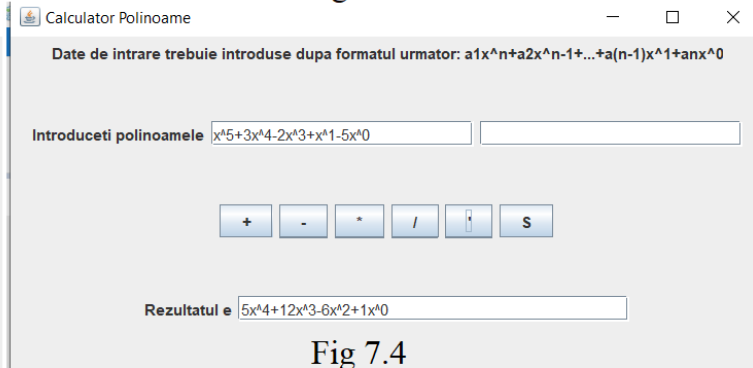


Fig 7.4

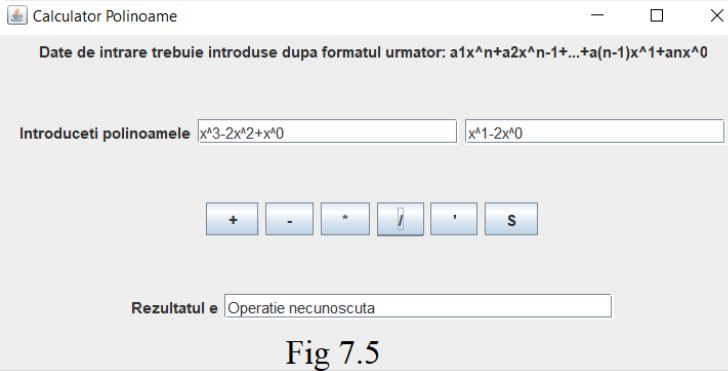


Fig 7.5

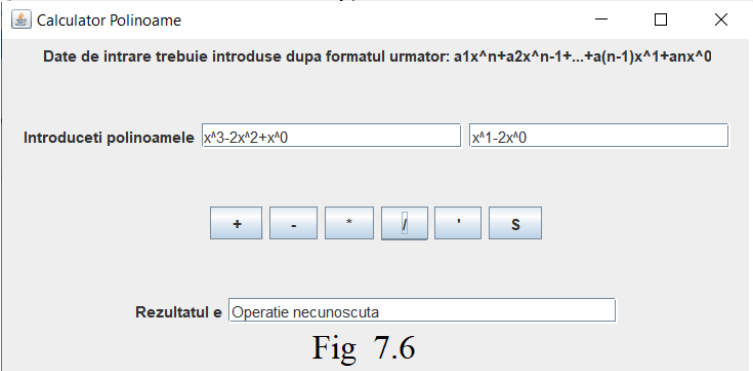


Fig 7.6

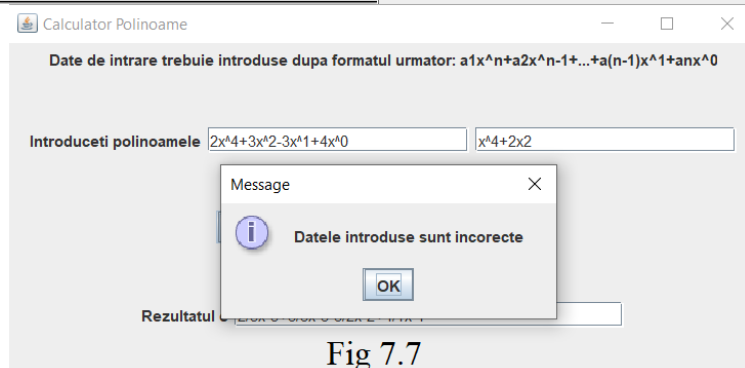


Fig 7.7

In figurile de mai sus sunt reprezentate cateva scenarii pentru a testa corectitudinea algoritmilor implementati pe care i-am utilizat in realizarea acestui proiect. In figurile 1, 2, 3, 4 sunt realizate operatiile matematice: adunare, scadere, inmultire, derivare in aceasta ordine pe niste polinoame aleator introduse pentru a verifica si totodata pentru a demonstra eligibilitatea rezultatelor din punct de vedere matematic furnizate de catre calculatorul realizat. In figura 7.5 am accesat butonul cu eticheta “/”, insa din cauza absentei unui algoritm de implementare al operatiei de impartire a doua polinoame in acest proiect am ales afisarea mesajului de eroare - “Operatie necunoscuta” . In ultima figura (Fig 7.6) este apasat butonul cu eticheta “S” (am ales litera “S” ca eticheta a acestui buton deoarece il consider cel mai apropiat caracter apartinand tastaturii de simbolul matematic specific integrarii) si este apelata functia de integrare, iar tehnica de afisare abordata se datoreaza mentinerii tipului variabilei de data (int) al coeficientului, varabila declarata in clasa Monom, caracteristica prezentata si mai sus (vezi Capitolul 6.3). In figura 7.7 este prezentat scenariul in care utilizatorul introduce datele gresit (la cel de al doilea polinom la al doilea termen lipseste caracterul “^”).In acest caz pe ecran va apare mesajul “Datele introduse sunt incorecte”.

8. Concluzii si posibilitati de dezvoltare ulterioare

In timpul dezvoltarii acestei teme, adica in timpul realizarii unui calculator de polinoame consider ca am reusit sa aprofundez atat unele paradigme ale programarii orientate pe obiect cat si abilitatile mele de a scrie cod si/ sau a corecta eventualele erori atat de sintaxa cat si la nivel de gandire aparute pe parcursul descrierii algoritmilor necesare functionalitatilor acestui aparat de calcul. Pe langa acestea am realizat cat de importanta este citirea cu atentie si intelegerea unei cerinte si capabilitatea de a structura/ imparti problema initiala in mai multe probleme mai mici si inceperea rezolvarii acestora ca un prim pas in indeplinirea sarcinilor unei teme,ca apoi sa asociezi toate rezultatele obtinute din surse pentru a ajunge la un rezultat corect in ansamblu. Am considerat utila aceasta tema si din prisma faptului ca mi-a solicitat cunostinte si din alte domenii si a promovat capacitatea auto-didactica a fiecarui student.

O posibilitate de dezvoltare a proiect realizat si descris in acesta documentatie ar putea fi:

- adaugarea unei metode care sa contina implementarea operatiei de impartire a doua polinoame.
- mai mult de atat s-ar putea face o revizuire asupra tuturor algoritmilor folositi si realizarea implementarii acestora intr-o maniera mai eficienta din punct de vedere a timpului de executie sau a memoriei folosite sau mai usor de descris (pentru programator) folosind functii deja definite ale pachetelor Java, astfel reusind o reducere a timpului petrecut pentru asigurarea functionarii unui astfel de aplicatii.
- o alta posibilitate de dezvoltare a proiectului ar putea fi imbunatatirea interfetei grafice apeland la partea artistica a fiecarui programator

9. Bibliografie

<https://www.tech-recipes.com/rx/39279/create-a-new-maven-project-in-eclipse/>

<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

https://www.w3schools.com/java/java_arraylist.asp