

Sistem de procesare a polinoamelor

Halasz Lorand Daniel

Grupa 30226

Cuprins

1. Obiectivul temei
2. Analiza si modelarea problemei
   1. Cazuri de utilizare si scenarii
   2. Asumptii
3. Proiectare
   1. Decizii de proiectare
   2. Diagrame UML
   3. Structuri de date
   4. Proiectare clase
   5. Pachete
   6. Interfata utilizator
   7. Modul de tratare a erorilor
4. Implementare
5. Testare
6. Rezultate
7. Concluzii
8. Bibliografie
9. Obiectivul temei

Polinoamele, impreuna cu setul de operatii, reprezinta un domeniu vast, studiat si utilizat in domeniul matematic si informatic. Obiectivul principal al temei consta in propunerea, proiectarea si implementarea unui sistem de procesare a polinoamelor de o singura variabila cu coeficienti intregi. Obiectivul principal poate fi descompus in mai multe obiective secundare, si anume:

* implementarea urmatoarelor operatii pe polinoame:
  + adunare
  + scadere
  + inmultire
  + impartire
  + derivare
  + integrare
* implementarea unei interfete grafice
* implementarea unei unitati de testare

1. Analiza si modelarea problemei

Analiza problemei consta in identificarea domeniului problemei. De cele mai multe ori, problema fiind una complexa urmeaza a fi descompusa in subprobleme pentru a ne usura munca. Aceste subprobleme pot fi, la randul lor, descompuse in alte subprobleme, acest pas repetandu-se pana in momentul in care se ajunge la un nivel de dificultate minim, care poate fi rezolvat. Urmatorul pas consta in identificarea domeniului solutiei. Acest lucru este realizabil prin aplicarea in repetate randuri a unui proces de abstractizare, pana cand se obtine ceea ce se doreste.

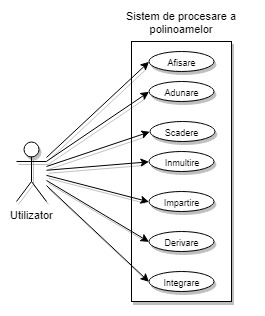
Odata analizate aceste lucruri se formeaza o idee generala asupra problemei. Urmatorul pas consta in descrierea cadrului de cerinte functionale, ceea ce se reduce la prezentarea modului in care sistemul reactioneaza in anumite situatii si la anumite intrari. In cazul de fata, pentru sistemul de procesare a polinoamelor se cunoaste faptul ca aplicatia trebuie sa realizeze anumite operatii (prezentate in capitolul 1) si sa ofere o interactiune mai usoara pentru utilizator, prin intermediul unei interfete grafice. Tot in cadrul analizei problemei se considera faptul ca un polinom este alcatuit din unul sau mai multe monoame. Scrierea polinoamelor se va face la fel ca si in toate sistemele de procesare a polinoamelor, si nu numai, si anume sub forma:

„”, unde sunt coeficinti intregi

Procesul de dezvoltare software se realizeaza pornind de la anumite nevoi. Acestea sunt translatate in cerinte, care urmeaza a fi validate de un client. Pe urma se concepe design-ul prin proiectarea de ansamblu si de detaliu. Se implementeaza componentele si se testeaza separat, dupa care acestea se integreaza in sistem, care la randul sau trebuie validat si testat ca intreg.

* 1. Cazuri de utilizare si scenarii

Cazurile de utilizare reprezinta un set de scenarii legate de modul in care este utilizat sistemul. Acestea ilustreaza modul in care un actor (in cazul de fata un utilizator) interactioneaza cu sistemul. Diagrama de utilizare (diagrama use-case) ne ajuta sa descoperim entitatile sitemului, atributele, actorii si comportamentul sistemului.

In afara cazului de utilizare „afisare”, in toate celelalte cazuri, utilizatorul va incepe prin a introduce cele 2 polinoame in campurile prestabilite. Dupa care va selecta operatia dorita (adunare, scadere, inmultire, impartire, derivare sau integrare). Rezultatul va fi afisat in partea de jos a interfetei. In cazul in care utilizatorul doreste sa verifice daca polinomul introdus este corect va apasa pe unul din cele doua buntoane de afisare, in functie de polinom pe care doreste sa il verifice. Introducerea polinoamelor se va face in format clasic, cu mici ajustari:

* x la puterea y, se va scrie x^y
* dupa coeficient nu se va introduce semnul de inmultire ’\*’, exemplu: 3x^2 – corect, iar 3\*x^2 – incorect
* nu se va lasa spatii in intre monoame, semne, coeficienti si exponenti
* nu este necesar ca polinomul sa fie introdus ordonat descrescator dupa exponenti

Introducerea datelor in mod corect reprezinta un best scenario. Utilizarea sitemului in mod corect presupune introducerea celor doua polinoame in formatul prezentat, dupa care este indicat sa se afiseze polinoamele pentru a putea vedea corectitudinea lor. Acestea vor fi afisate ordonate descrescator dupa exponent. Ex: „x^3+3x^2-x+5”. Totusi, acest lucru nu este sigur si ne putem astepta la diferite lucruri, dintre care:

* introducerea unor caractere nespecifice introducerii unor polinoame, precum :’&’, ’#’, ’%’...
* introducerea unui text care nu reprezinta un polinom
* introducerea unor spatii sau a semnului de inmultire ’\*’
* introducerea unor exponenti negativi sau a unor coeficienti cu virgula
  1. Asumptii

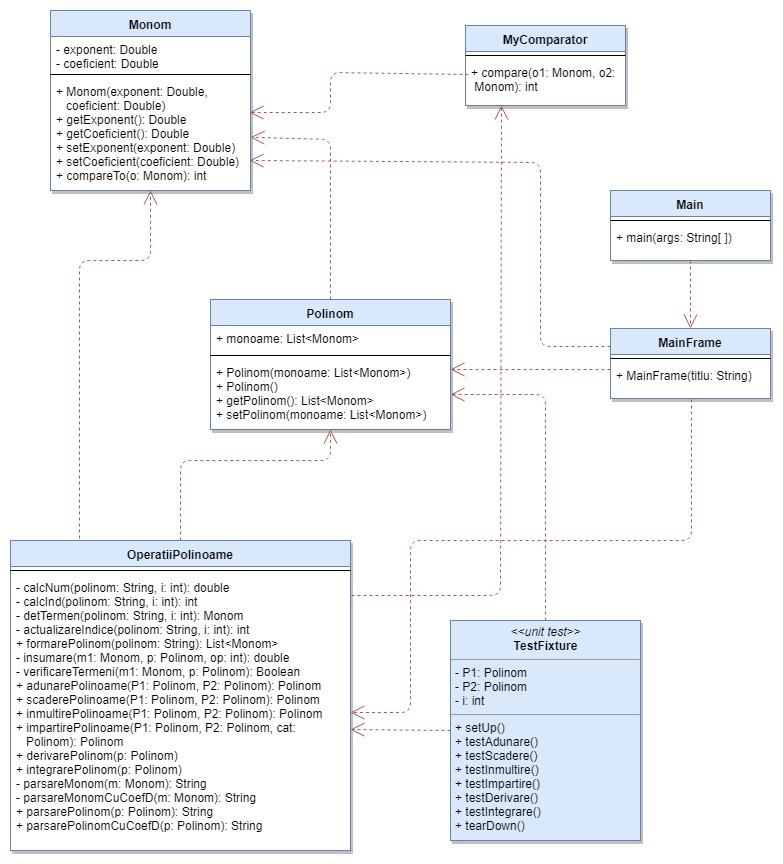
Asumptiile stabilesc cadrul in care se dezvolta aplicatia. In cadrul proiectului s-a stabilit urmatoarea asumptie: presupunem ca datele de intrare, adica cele 2 polinoame, sunt introduse in mod corect, conform specificatiilor anterioare, nefiind necesara o ordonare prealabila a acestora. In schimb cazurile de impartire cu 0, derivare sau integrarea cifrei 0 sau introducerea unui polinom in care avem mai multe monoame cu acelasi exponent (de exemplu: „x^2+x+x^3+x^2+x-x+2”) sunt rezolvate.

1. Proiectare
   1. Decizii de proiectare

Pornind de la definitia de baza a polinomului, am luat decizia de descompunere a polinomului in doua clase. Prima, si clasa de baza de la care se porneste, este clasa Monom, care retine coeficientul si exponentul lui ’x’. Cea de-a doua clasa este clasa Polinom care retine o lista de monoame. De asemenea am decis ca intrarile sistemul, adica cele 2 polinoame, sa codifice string-uri, care ulterior, in cadrul sistemului sunt separate si transformate in polinoame.

* 1. Diagrama UML

Urmatoarea diagrama UML descrie clasele si obiectele sistemului si relatiile dintre acestea:



* 1. Structuri de date

Pentru definirea polinomului se utilizeaza o lista de monoame. Acest lucru este realizabil utilizand colectii, implicit clasa ArrayList<T> care implementeaza toate metodele din interfata List<T>.

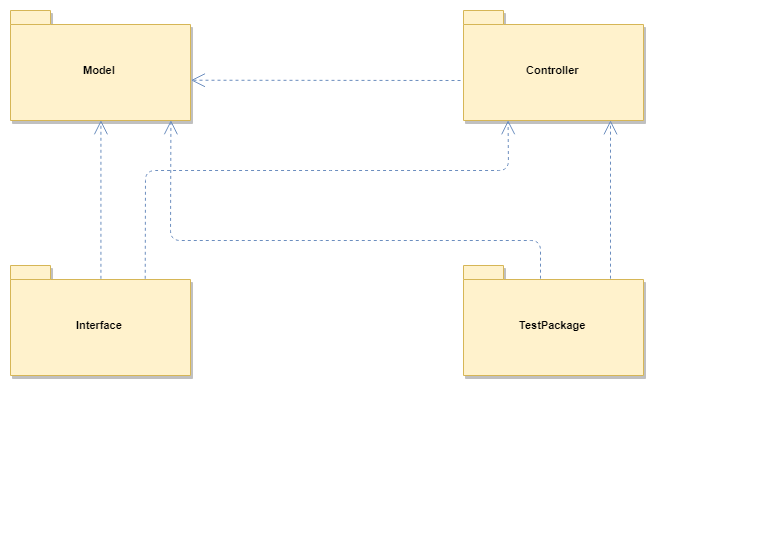
* 1. Proiectare clase

Sistemul de procesare a polinoamelor contine 7 clase:

1. Clasa **Monom** – clasa de baza a proiectului ce defineste structura de monom, care contine coeficientul si exponentul acestuia. Aceasta mai contine metoda compareTo care sta la baza sortarii polinoamelor dupa exponent.
2. Clasa **Polinom** – clasa care defineste structura de polinom ca o lista de monoame
3. Clasa **MyComparator** – clasa care implementeaza interfata Comparator si care contine metoda cu ajutorul careia, impreuna cu metoda compareTo din Monom, se sorteaza lista de monoame din cadrul clasei Polinom.
4. Clasa **OperatiiPolinoame** – clasa care contine toate operatiile care se realizeaza pe polinoame, incepand de la formarea polinoamelor, la parsarea lor in string-uri pentru scrierea rezultatului si inclusiv operatiile de adunare, scadere, inmultire, impartire, derivare si integrare a polinoamelor.
5. Clasa **MainFrame** – clasa care sta la baza interfetei grafice si care practic construieste un mod mai usor de interactionare a utilizatorului cu sistemul.
6. Clasa **Main** – clasa care instantiaza interfata grafica, acest lucru ducand la „punerea in miscare” a intregului sistem
7. Clasa **TestFixture** – clasa de test prin care se verifica corectitudinea operatiilor
   1. Pachete

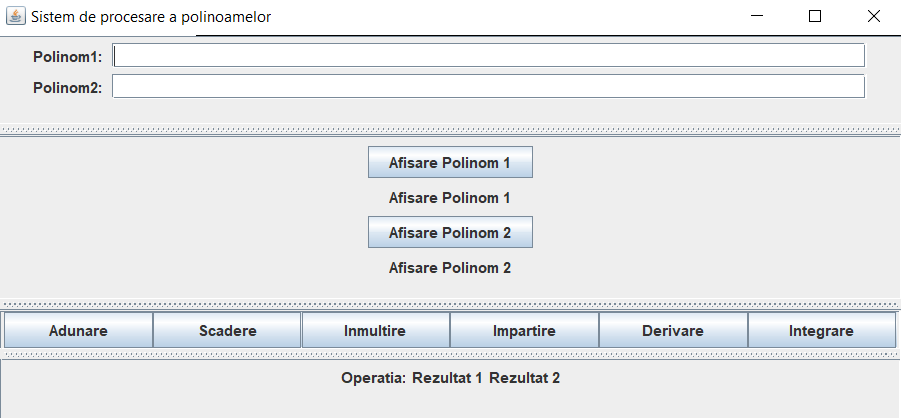
Diagramele de pachet ne ajuta la descompunerea sistemelor mari in subsisteme, lucru necesar pentru a putea realiza o implementare sistematica. Sistemul de procesare a polinoamelor este impartit in 4 pachete:

1. Pachetul Model – contine clasele Monom, Polinom si MyComparator.
2. Pachetul Controller – contine clasa OperatiiPolinoame
3. Pachetul Interface – contine clasele Main si MainFrame
4. Pachetul TestPackage – contine clasa de testare



* 1. Interfata utilizator

Interfata grafica reprezinta un mod mai usor de interactionare dintre utilizator si program. Fara aceasta, utilizarea sistemului de procesare a polinoamelor devine mai greoaie si destul de dificila pentru un utilizator nespecializat. Interfata este realizata cu ajutorul clasei MainFrame care extinde clasa JFrame.



Interfata este compusa dintr-o fereastra cu titlul „Sistem de procesare a polinoamelor” si care este impartita in 4 zone, astfel:

1. Zona 1 – zona compusa din doua label-uri si doua textfield-uri in care utilizatorul va introduce polinoamele dorite
2. Zona 2 – zona utilizata pentru a vedea polinoamele introduse in mod corect, acestea fiind sortate descrescator dupa exponent
3. Zona 3 – zona compusa din mai multe butoane si care are rolul de a permite utilizatorului sa aleaga operatia dorita
4. Zona 4 – zona in care utilizatorul poate vedea rezultatul operatiilor alese
   1. Modul de tratare a erorilor

Ca in orice sistem, aparitia unor erori este foarte posibila si trebuie luata in calcul. In cazul de fata o eroare destul de posibila este impartirea cu primului polinom cu polinomul nul (adica impartirea cu 0). Acesta eroarea a fost tratata special in cadrul programului prin verificarea celui de-al doilea polinom, iar in cazul in care acesta ar fi 0, sistemul va raspunde cu un mesaj de specific care va fi afisat in interfata grafica in zona de rezultate. O alta eroarea destul de posibila consta in aparitia in cadrul aceluiasi polinom a unor monoame cu acelasi exponent (de exemplu ”x^2+5x^3+x^2-2x+6+x”), eroare este tratata prin insumarea coeficientilor monoamelor cu acelasi exponent.

1. Implementare

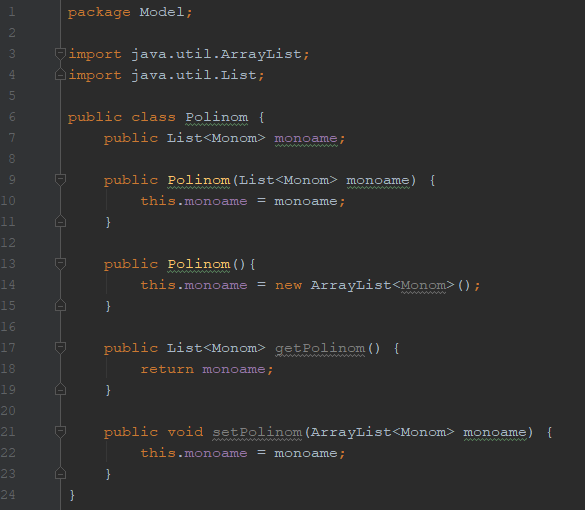
Dupa stabilirea obiectivului principal si a obiectivelor secundare si dupa analiza, modelarea si proiectarea proiectului urmeaza implementarea lui.

1. Clasa **Monom**



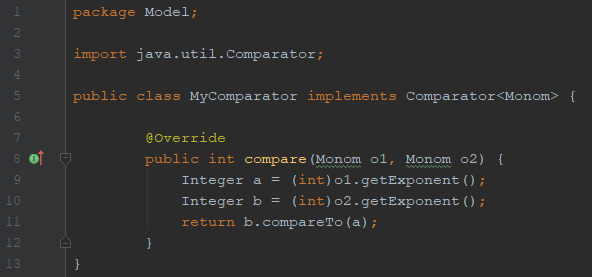
Clasa Monom, care implementeaza interfata Comparable, are ca si variabile de instanta exponentul si coeficientul monomului. In cadrul acestei clase este descris constructorul clasei, getterii, setterii si metoda compareTo, implementata pentru a putea realiza, ulterior, ordonarea descrescatoare a polinoamelor dupa exponent. Aceasta este o clasa de baza deoarece monomul reprezinta cea mai mica unitatea care formeaza un polinom si astfel ajuta la descompunerea problemei in subprobleme.

1. Clasa **Polinom**



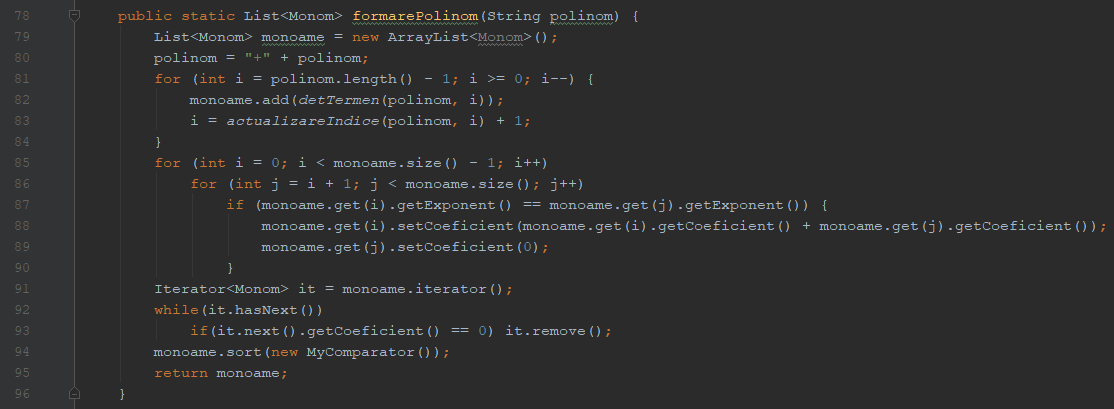
Clasa polinom are ca variabila de instanta o lista de monoame. In cadrul clasei se definesc doi constructori, dintre care unul cu parametrii, pentru crearea unui polinom avand o lista de monoame, iar cel fara parametrii pentru crearea unui polinom gol care urmeaza a fi construit pe parcurs. Ca si in cazul precedent, clasa Polinom mai contine un getter si un setter.

1. Clasa **MyComparator**

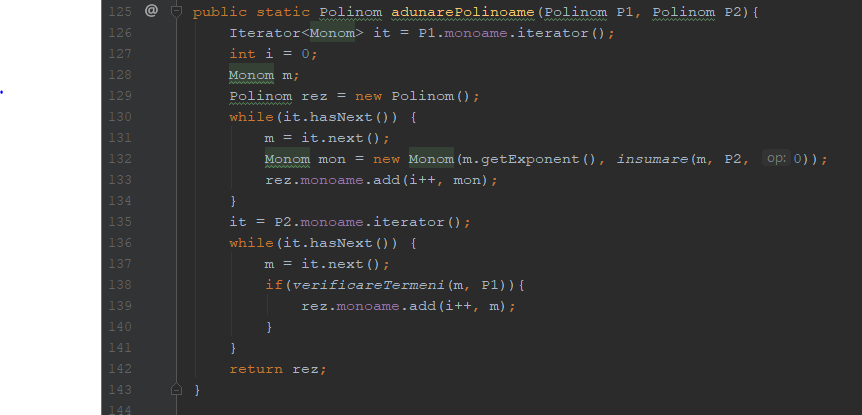


Clasa MyComparator implementeaza interfata Comparator si suprascrie metoda compare pentru a putea sorta descrescator polinomul dupa exponent, lucru necesar atat din punct de vedere estetic, cat si din punct de vedere al implementarii, deoarece, de exemplu in cazul impartirii avem nevoie ca cele doua polinoame sa fie sortate pentru a putea stii care este monomul de grad maxim.

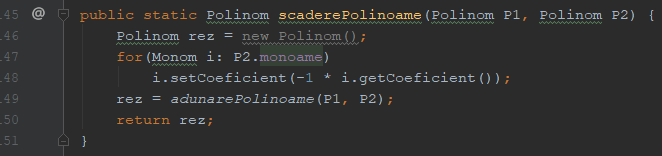
1. Clasa **OperatiiPolinoame**

In clasa OperatiiPolinoame se afla toate operatiile care se fac cu polinoame: adunare, scadere, inmultire, impartire, derivare si integrare. De asemenea in cadrul clasei se mai descriu alte doua operatii semnificative pentru rezolvarea problemei, si anume operatia de convertire dintr-un string in polinom si operatia de parsare pentru a putea afisa polinomul sub forma unui string.

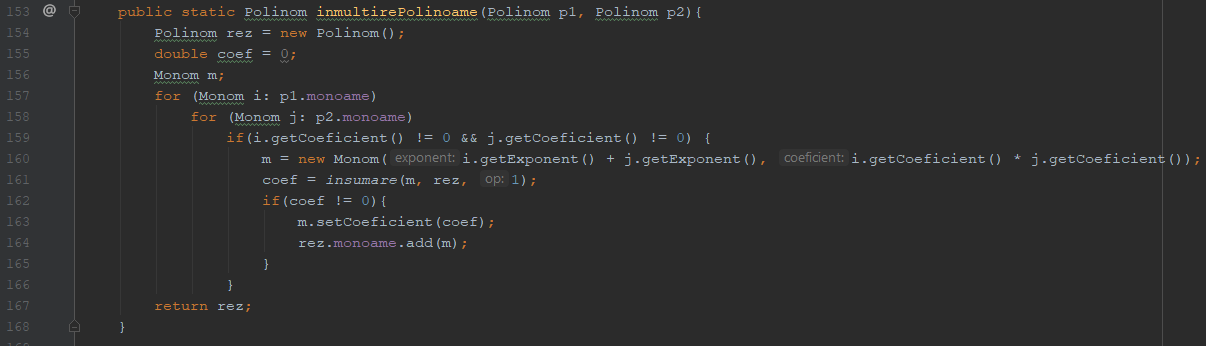
Aceasta este metoda principala de convertire a unui String intr-un polinom. Initial se declara o lista de monoame, iar String-ului primit ca si parametru i se concateneaza la inceput semnul ‘+’ pentru a putea determina corect coeficientul primului termen din polinom. Se incepe construirea polinomului de la dreapta spre stanga. Dupa determinarea polinomui se verifica daca nu exista mai multe monoame cu acelasi exponent (ex: “x^2+x+2x^2-3x+8”). In cazul in care acestea exista, din ele se face un singur monom, adunandu-se coeficienti cu acelasi exponent. Odata parcurs, in momentul in care se gasesc acesti termeni, se seteaza coeficientul celui de-al doilea monom la 0. La final se elimina toate monoamele cu care au coeficientul 0.

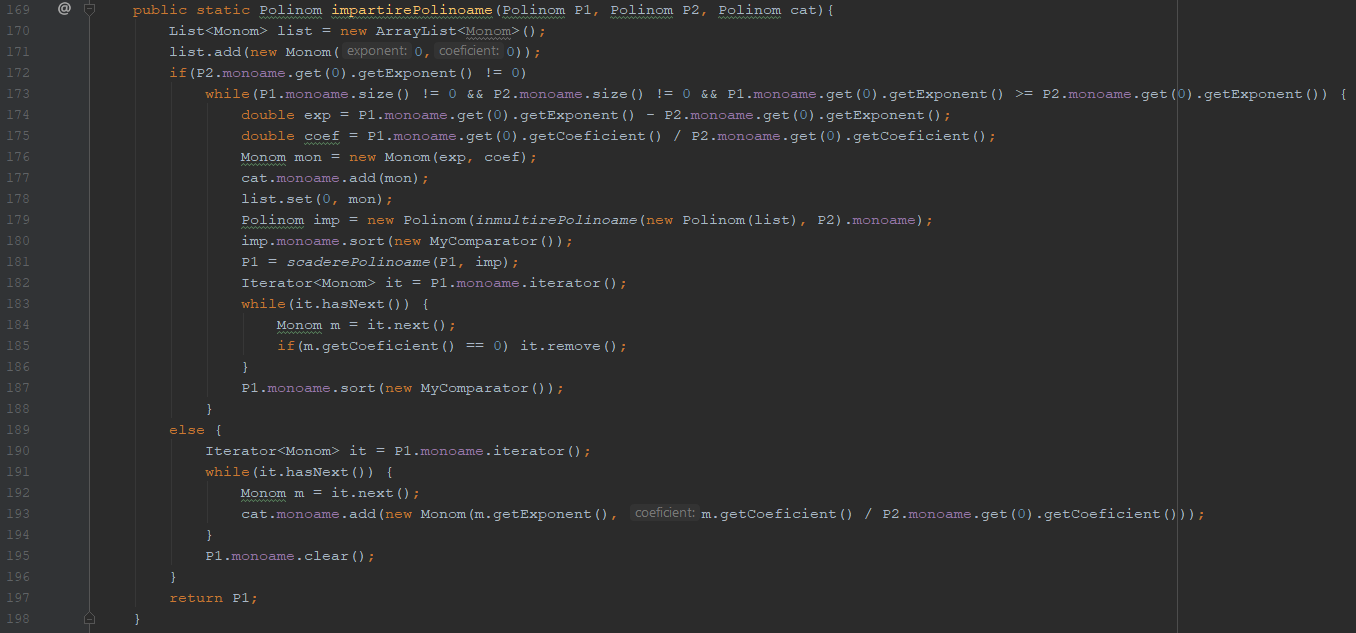


Metoda adunarePolinoame functioneaza pe urmatorul principiu: se parcurge prima lista si se aduna coeficientii corespunzatori din polinomul 2, dupa care se adauga toate elementele ramase in polinomul 2.



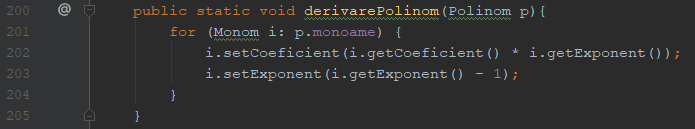
Metoda de scadere a polinoamelor se realizeaza ca o adunare, doar ca se inmultesc, in prealabil, coeficientii celui de-al doilea polinom cu -1.

Inmultirea a doua polinoame se face efectiv prin inmultirea fiecarui monom din primul polinom cu fiecare monom din cel de-al doilea polinom. Inmultirea consta in adunarea exponentilor si inmultirea coeficientilor. Dupa inmultirea a doi termeni se verifica daca mai exista termeni cu acelasi coeficient in rezultat, iar in caz afirmativ acestia se vor aduna.

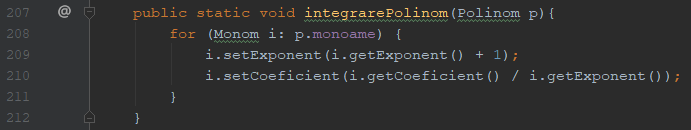


Impartirea a doua polinoame se face identic cu modul in care se impart polinoamele pe hartie, si anume:

* se imparte primul monom din deimpartit la primul monom din impartitor (se scad exponentii)
* monomul rezultat se pune in cat
* se inmulteste impartitorul cu monomul rezultat in urma impartirii si se pune in polinomul imp
* se face diferenta dintre deimpartit si polinomul imp
* se repeta pasii de mai sus pana ce gradul impartitorului este mai mare decat gradul deimpartitului
* deimpartitul devine rest



Derivarea unui polinom se face prin inmultirea coeficientului cu exponentul si prin decrementarea exponentului.



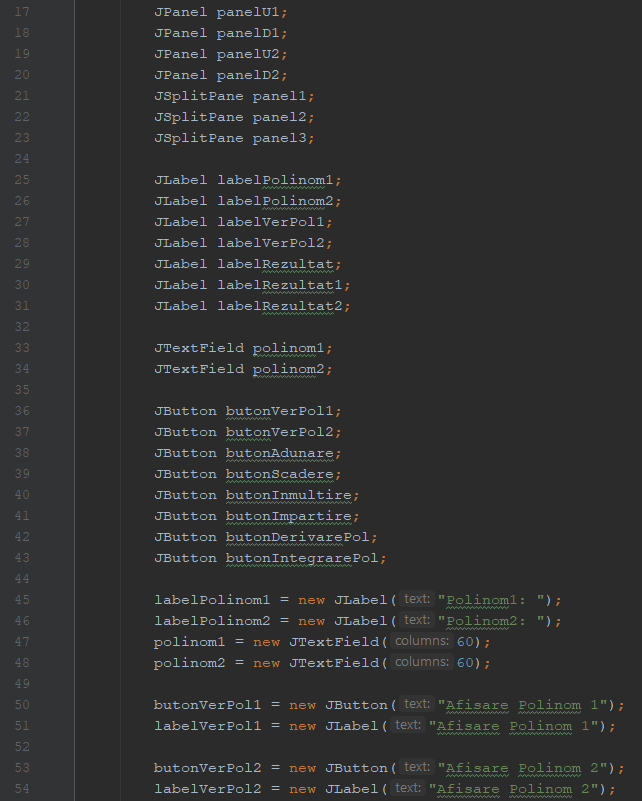
Integrarea unui polinom se face prin incrementarea exponentului si prin impartirea coeficientului la noul exponent.

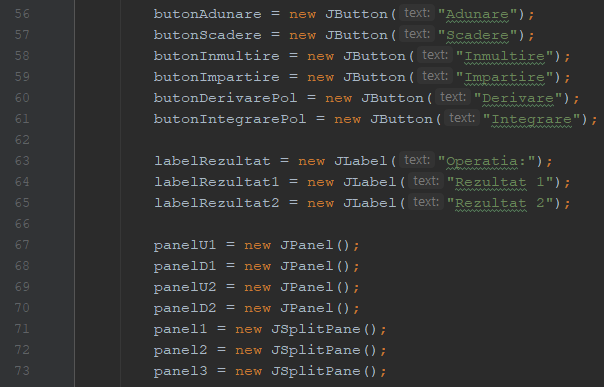
Tot in clasa OperatiiPolinoame mai avem 2 metode parsare a polinoamelor, una care se foloseste in cazul operatiilor de adunare, scadere, inmultire, impartire si derivare (coeficientii monoamelor raman intregi) si alta care se foloseste in cazul integrarii, deoarece, aici, in urma impartirii coeficientului la exponent pot sa apara coeficienti cu virgula, caz care trebuie tratat separat.

In afara de metodele prezentate mai sus, mai exista cateva metode necesare pentru a putea face toate operatiile in mod corect.

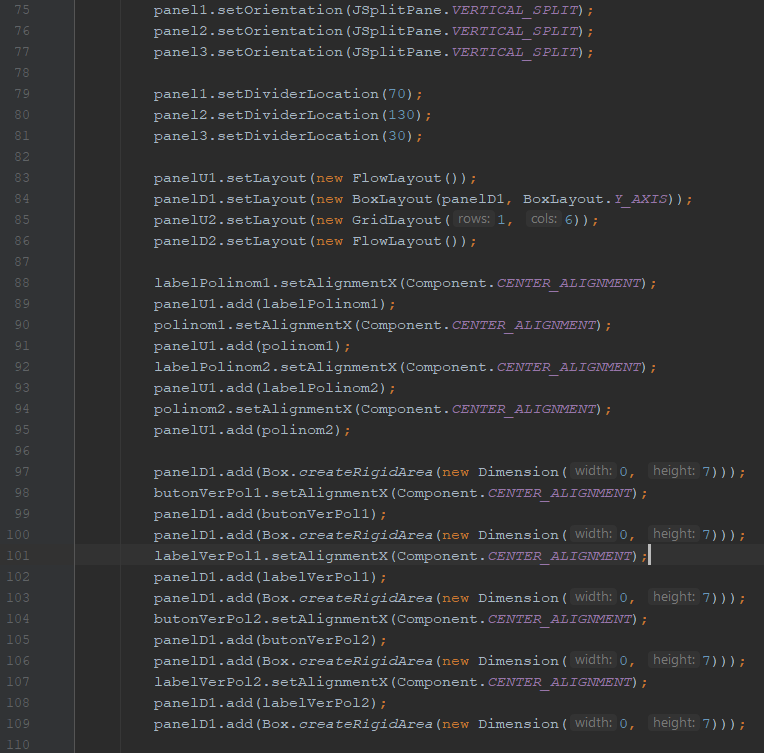
1. Clasa MainFrame

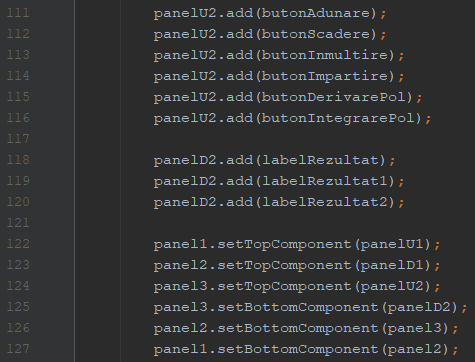
Clasa MainFrame realizeaza practic toata interfata grafica. Aceasta clasa extinde clasa JFrame si contine un singur constructor, care primeste ca parametru un String reprezentand titlul, iar mai apoi apeleaza superconstructorul caruia ii da ca parametru acest String. Dupa care urmeaza o serie de declarare de componente si mai apoi instantierea lor.



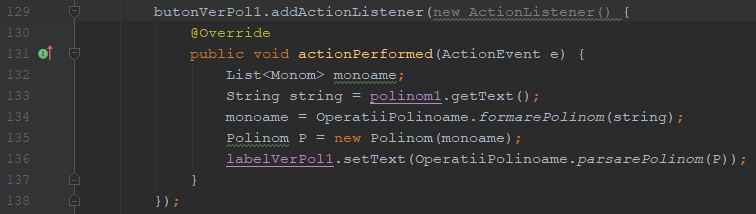


Mai apoi urmeaza efectiv crearea interfetei grafice prin stabilirea layout-ului si prin adaugarea componentelor (butoane, textfield-uri, label-uri)

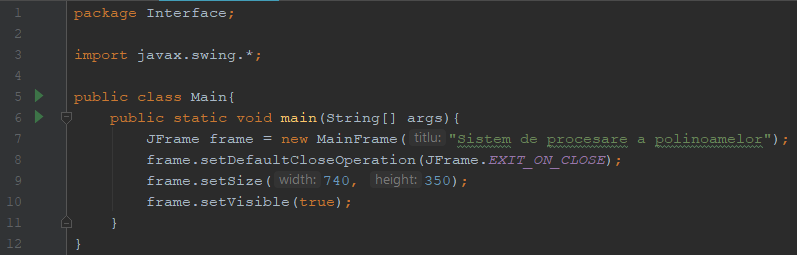




Dupa care, pentru fiecare buton in parte, se creeaza cate un ascultator (Listeners), care se apeleaza in momentul in care utilizatorul interactioneaza cu interfata, provocand astfel un eveniment

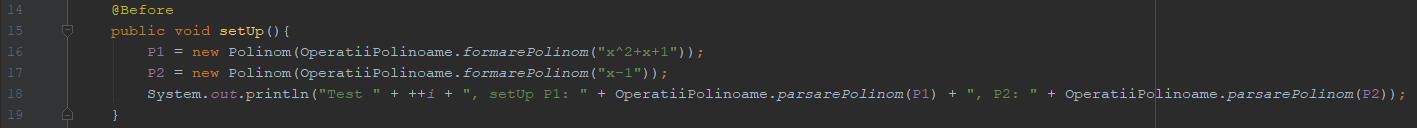


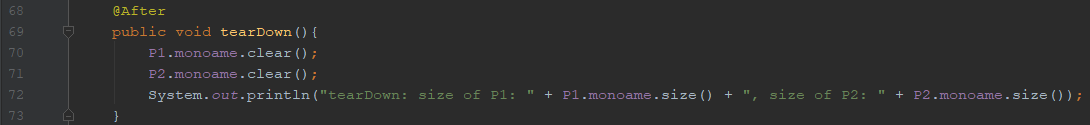
1. Clasa Main

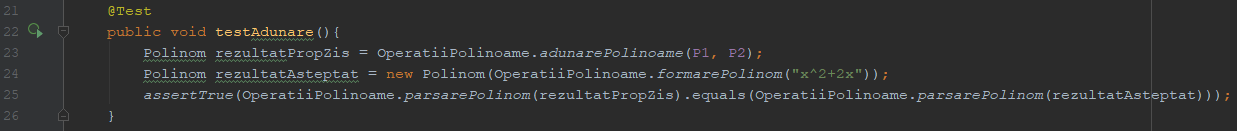


In interiorul clasei Main se creaza un obiect de tipul MainFrame, caruia i se seteaza titlul si dimensiunea si astfel se deschide interfata.

1. Clasa TestFixture

Pentru partea de testare s-a realizat o clasa separat, si anume clasa TestFixture, in interiorul careia fiecare test are pregatit ca punct de plecare in momentul rularii o valoare prestabilita. Acest lucru s-a realizat cu ajutorul metodelor setUp(), metoda ce este apelata automat inainte de fiecare test, si tearDown(), metoda ce este apelata automat dupa fiecare test. In afara celor 2 metode s-au mai creat cate o metoda pentru fiecare operatie in parte.



 In continuare se va prezenta metoda de testare a operatiei de adunate:

Se formeaza un prim polinom, care este rezultatul adunarii dintre cele 2 polinoame setate la inceput, dupa care se formeaza polinomul pe care ne asteptam sa il obtinem prin adunarea celor 2 polinoame, iar in final le comparam.

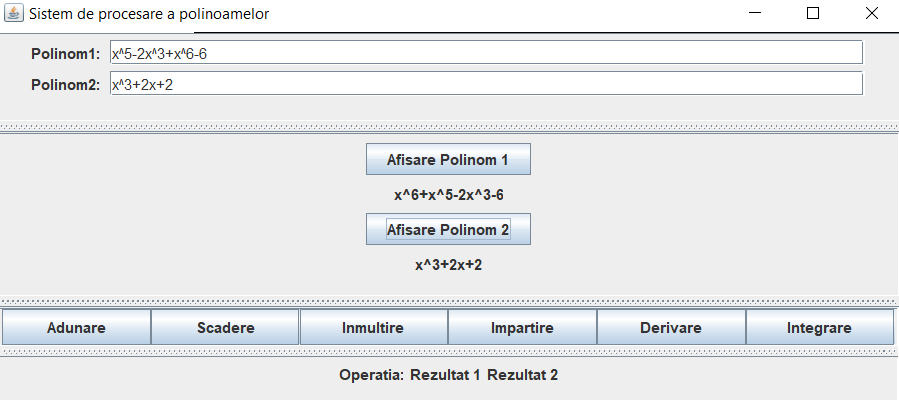
1. Testare

Partea de testare este analizata mai amanuntit in capitolul anterior la implementare.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ce se testeaza | Date de intrare pentru test | Rezultat asteptat | Rezultat efectiv obtinut | Pass/fail |
| Adunarea | P1 = x^2+x+1  P2 = x-1 | x^2+2x | x^2+2x | Pass |
| Scadere | P1 = x^2+x+1  P2 = x-1 | x^2+2 | x^2+2 | Pass |
| Inmultire | P1 = x^2+x+1  P2 = x-1 | x^3-1 | x^3-1 | Pass |
| Impartire | P1 = x^2+x+1  P2 = x-1 | Cat = x+2  Rest = 3 | Cat = x+2  Rest = 3 | Pass |
| Derivare | P1 = x^2+x+1  P2 = x-1 | P1 = 2x+1  P2 = 1 | P1 = 2x+1  P2 = 1 | Pass |
| Integrare | P1 = x^2+x+1  P2 = x-1 | P1 = 0.333333x + 0.5 x^2+x  P2 = 0.5x^2 -x | P1 = 0.333333x + 0.5 x^2+x  P2 = 0.5x^2 -x | Pass |

1. Rezultate

Dupa introducerea a doua polinoame, se apasa pe butoanele de afisare pentru a vedea polinoamele introduse, in ordine descrescatoare, dupa exponentul acestor.



Dupa care se selecteaza operatia dorita.

* Adunare



* Scadere



* Inmultire



* Impartire



* Derivare



* Integrare



1. Concluzii

O concluzie foarte importanta, a acestui proiect, consta in importanta descompunerii problemei in subprobleme si usurinta cu care o problema poate fi realizata in acest mod. De asemenea este foarte importanta realizarea unei analize amanuntite si formarea unei idei generale asupra problemei inaintea de a incepe implementarea ei.

Din acest proiect am invatat lucruri precum:

* Utilizarea colectiilor intr-un cadru mai amanuntit
* Imbogatirea cunostintelor in materie de interfata grafica
* Importanta utilizarii proceselor de modularizare, abstractizare si incapsulare
* Complexitatea realizarii unui proiect de la 0 care, pe langa partea de rezolvarea a problemei, necesita de asemenea o interfata grafica, o unitate de testare si o documentatie laborioasa a intregului proiect

Sistemul de procesare a polinoamelor are o gama larga de posibilitati de dezvoltare ulterioara, dintre care amintim:

* Posibilitatea de a efectua operatii cu polinoame ai caror coeficienti nu sunt doar intregi
* Cresterea numarului de operatii posibile prin adaugarea de noi operatii posibile, precum aflarea radacinilor sau descompunerea polinomului in produs de alte polinoame (in cazul in care acest lucru este posibil)
* Dezvoltarea interfetei grafice

1. Bibliografie

<http://www.coned.utcluj.ro/~salomie/PT_Lic/>

<http://users.utcluj.ro/~igiosan/teaching_poo.html>

<https://www.draw.io/>

<https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>