Descriere

Proiectul realizat are ca pornire cerinta de a realiza un sistem de proceasare a polinoamelor de o singura variabila cu coeficienti intregi. Operatile care au fost implementate sunt cele de adunare, scadere, inmultire, impartire, derivare si integrare. De asemenea s-a folosit o fereastra grafica pentru usurarea vizualizarii si introducerii datelor de intrare, fereastra care permite alegerea operatiei dorite, preluarea polinoamelor de intrare si afisarea rezultatului prin procesarea lor spre obtinerea rezultatului dorit. Tema a fost realizate urmand tiparul Model-View-Controller (MVC) si s-a respectat paradigma claselor si divizarii functilor astfel incat sa existe o clasa care se ocupa de partea vizuala (PolView), una care sa modeleze operatile care trebuie facute (PolModel) , o clasa (PolController) care sa lege partea vizuala de ce algoritmica si o clasa PolMain care instantiaza celelalte clase pentru obtinerea rezultatului dorit. Pe langa aceste clase proiectul mai contine o clasa Monom care are rolul de a pastra monoame, o clasa Polinom care incorporeaza un ArrayList de monoame si implementeaza operatile : add, sub, mul, iMulQ, iMulR, derivate, integrate, addString, si o clasa pentru testarea operatilor numita PolTest. Totul a fost creat folosinduse mediul InteliJ Idea (2016.3.5) , varianta java jdk1.8.0_112 si librariile “javax.swing”, “java.awt” componente din “java.util” si “junit” pentru testare.

Componente

PolMain

PolMain este clasa principala si contine metoda “main” care este apelata la rularea programului. Contine doar trei instantieri: o instanta mod pentru clasa PolModel alaturi de care se creaza un obiect nou la executarea constructorului fara parameti a acesteia, o instanta view pentru clasa PolView care creaza un obiect prin constructor si un obiect “afisare”, instanta a clasei PolController al carui constructor primeste obiectele “view” si “mod” drept parametri.

Singeorzan Ionut-Florin
Gr: 30227

PolView

Aceasta clasa se ocupa strict de partea vizuala a proiectului, de asezarea containarelor de setarea valorilor lor, de preluarea datelor de intrare si de setarea listeneror. PolView extinde clasa JFrame pentru generarea ferestrei grafice la instantierea unui obiect de tipul clasei. Toate obiectele continute sun declarate private pentru a nu putea fi apelate din exteriorul clasei iar obiectele de tipul JPanel si CardLayout au si specificatorul static. Obiectele declarate sunt: un container CardLayout pentru a putea modifica panoul actual; 8 JPanel-uri, unul care le contine pe toate “winPanel”, unul pentru meniul de selectie menuPanel si inca 6, cu denumirile op1... op6 pentru fiecare operatie; un obiect “choose” de tipul JComboBox care presupune alegerea actiunii dorite; doua tablouri JTextField-uri “polinom1” si “polinom2” de cate 6 elemente fiecare; 6 elemente Jlabel pentru rezultat “polResult” , unul siplimentar pentru operatia de impartire “polResult2”, inca un tablot text de 6 elemente pentru un text afisat la fiecare operatie si un “textMenu” pentru afisarea unui text in meniu; doua tablouri JButton “backButton” si “calc” de 6 elemente si un obiect de tipul int, “state” care e initializat cu 0 si are rolul de a retine actiunea aleasa.

Clasa are 18 metode care dau feresrei un aspect organizat:

- Metoda setFrame da specificati necesare ferestrei precum dimensiune, titlu si vizibilite;
- Metode “init()” initializeaza obiectele create la valorile dorite;
- Metoda “addBounds”plaseaza la diferite coordonate obiectele create si seteaza spatiul ocupat de acestea in fereastra;
- Metoda “menu()” seteza containarele ce o sa apara in fereastra grafica si adauga iteme obiectului “choose”;
- “addComponents” are rolul de a plasa tabloul “text” de a atasa prin metoda “addToPanel”(metoda de atasare a panoului curent) fiecare componenta in panoul ei. Ea apeleaza metodele “menu” si “addBounds”;
- “incPanel” seteaza toate panourile pe panoul winPanel, daundule constrangeri de nume cifrele 0, 1, .., 7;

- Constructorul fara nici un parametru PolView apeleaza metodele de mai sus, adauga winPanel ferestrei, seteaza Layout-urile null si culoarea alba pentru fiecare panou;
- Metodele getSelected, getFirstPol, getSecondPol, getState, setPolResult, setState, setPolResult2 au rolul de a prelua sau seta obiectele corespunzatoare la valoarea parametrilor;
- “attachBack”, “attachCalc” si “attachChoose” primesc ca parametru un ActionListener si adauga actiuni obiectelor aferente.

Monom

Contine trei variabile: coef(coeficientul intreg al unui monom), coefDouble(coeficientul real) si deg(gradul unui monom). Clasa are trei constructori, unul ce seteaza paramatri interni la valorile primite si cel real la 0.0, unul care seteaza coeficientul intreg la 0 si ceilalti doi la parametri actuali primiti si inca un constructor care atribuie identificatorilor monomului valorile nule(zero). Din clasa mai fac parte “gettere si settere” care preiau sau modifica variabilele monomului. Pentru sortarea monoamelor in polinom clasa implementeaza interfata “Comparable” si suprascrie metoda compareTo pentru aranjarea descrescatoare a monoamelor in polinom.

Polinom

Aceasta clasa este proiectata sa retina un ArrayList de monoame denumit “polinom”. Ea trebuie sa impementeze metode pentru adaugarea unui element de tipul monom in arrayList si pentru afisarea unui polinom sub forma de string. Metoda “exist” primeste un argument de tipul int si verifica daca in polinom mai exista un monom cu gradul egal cu argumentul si in cazul true returneaza pozitia acestuia in polinom altfel returneaza valoarea -1. Exista doua metode “adauga” cu rolul de a adauga un nou monom polinomului prin specificarea coeficientului si gradului. Singura diferenta dintre cele doua este ca una adauga un coeficient real iar cealalta unul intreg. La inceput se verifica daca coeficientul care se doreste a fi adaugat e diferit de 0 si in cazul in care in polinom mai exista un monom cu gradul

dat ca argument, acestuia i se aduna argumentul iar in caz contrar polinomului i se adauga un nou monom -> polinom.add(new Monom(co,gr)). La final se face sortarea polinomului descrescator dupa grade prin apelarea metodei Colections.sort(polinom). Metoda afisare are ca obiectiv crearea si returnarea unui string care detine sub forma de sir inlantuirea monoamelor din polinom. Stringul de returnat se creaza in variabila “show” asadar initial se parcurg (cu k) elementele polinomului si se verifica pentru fiecare monom daca are coeficientul (real sau intreg) diferit de zero. Se foloseste formatul unui polinom: show=show+”+”{auxiliar}+k.getCoef()+”x^”+k.getDeg() si pentru coeficient real se utilizeaza la afisare doar doua zecimale. Cele doua ramuri ale clauzei fiecarui if verifica daca suplimentar trebuie adaugat semnul “+”, depinzand de semnul coeficientului si pozitia in polinim. Metoda “equals” verifica daca doua polinoame sunt egale. Ea este folosita la testare.

PolModel

PolModel reprezinta partea algoritmica a proiectului, ceea ce se intampla in spatele ferestrei grafice. Metodele de aici sunt declarate publice deoarece trebuie acesate din partea de control (mai putin doua care sunt utilizate doar in interiorul clasei). Metodele constituente sunt:

- Metoda “addString” are scopul de a prelucra un string pentru transpunerea lui intr-un obiect de tipul Polinom. Metoda returneaza “toReturn”, polinomul creat in urma operatilor pe sir. String-ul regex e format din trei parti: ”([\\-]?\\d+)”, grupul “1” care identifica coeficientul intr-un monom (ax^b), (x\\^), grupul “2” pentru “x^”, el nefiind folosit si grupul ”3” (\\d+) pentru exponent. Se creaza obiectele “patt” si “mat” pentru compilarea sirului “regex” si verificarea tiparului sirului de intrare. Sirul este prelucrat doar daca respecta forma unui polinom /*if (str.matches("(^(([\\-\\+]?\\d+)(x\\^)(\\d+))*$)"))*/ si se separa grupurile adauganduse in polinomul “toReturn” coeficientul si gradul fiecarui monom in parte;
- Metoda “isEmpty” returneaza fals daca arrayList-ul de monoame contine cel putin un monom altfel se returneaza true;

- “add” este o metoda care aduna doua polinoame si returneaza polinomul rezultat. Ea adauga polinomului de returnat “res” primul si cel de al doilea polinom. Adunarea este efectuata datorita faptului ca, in metoda adauga(); atunci cand este introdus un nou monom, in cazul in care gradul mai exista coeficientii se aduna;
- Metoda “sub” returneaza un polinom in care se face scaderea polinoamelor ca parametri de intrare. Lui i se adauga primul polinom si folosind aceeasi metoda “adauga” se pune in el inversul coeficientului celui de al doilea. Metoda verifica daca acest coeficient este real sau intreg si modeleaza astfel rezultatul scaderii;
- Metoda de imultire “mul” returneaza produsul parametrilor de intrare sub forma de polinom. Se parcurg element cu element monoamele celor doua polinoame, se preia doar coeficientul care e diferit de 0 si se adauga polinomului de returnat produsul coeficientilor fiecarui monom si suma gradelor lor. Polinomul returnat va avea coeficinti retinuti in coefDouble.
- “derivate” este metoda care prelucreaza un polinom si returneaza polinomul derivat (functie de x). Se parcurg monoamele si se adauga in polinomul de returnat produsul dintre coeficient si grad si gradul minus 1 /*resDerivate.adauga(i.getCoef()*i.getDeg(),i.getDeg()-1) */;
- Metoda opusa derivarii, “integrate” returneaza monomul format prin adaugarea (in coefDouble) rezultatul impartii dintr coeficientul intreg si grad, gradul se incrementeaza;
- Metoda “getMonomMax” are rolul de a extrage dintr-un polinom monomul de grad maxim. Se parcurge polinomul element cu element, se verifica daca monomul are un coeficient diferit de zero, se preia coeficientul, se compara cu o variabila degMax unde se pastreaza gradul maxim si se formeaza monomul prin adaugarea gradului si coeficientului. Monomul rezultat se returneaza;
- Metoda “getDouble” parcurge monom cu monom un polinom si translateaza coeficientul intreg la coeficient real. Se returneaza polinomul cu coeficienti reali;
- Impartirea este separata in doua metode: iMulQ si iMulR, pentru returnarea catului si restului. “iMulQ” respecta shema clasica de impartire a

polinoamelor. In “aux” se pastreaza polinomul curent, initial egal cu polinomul care trebuie impartit. “aux2” reprezinta modificarea lui “aux” pas cu pas prin scadere iar “aux3” pastreaza catul temporar. Atata timp cat dimensiunea auxiliarului de impartit nu depaseste variabila de test index si cat timp monomul cu grad maxim si coeficient diferit de zero are gradul mai mare decat gradul monomului cu index 0 se efectueaza algoritmul matematic de impartire. Se ia coeficientul diferit de zero al polinomului cu care trebuie sa impartim si il impartim la coeficientul auxiliarului cu monomul de grad maxim. Se adauga catului Q acest coeficient obtinut si diferenta gradelor din cele doua poliname (acestea se adauga si lui aux3). Se face imultirea dinte polinomul cu care trebuie sa impartim si “aux3”, rezultatul este scazut din aux si aux3 se scade cu el insusi. La iesirea din while Q va retine catul impartirii si va fi returnat.”iMulR” va face operatiile inverse (de verificare) pentru aflarea restului. Matematic, se afla catul, se imulteste catul cu polinomul la care se imparte si se scade din polinomul de impartit rezultatul care se va returna. Daca gradele difera(mai mare la impartitor) se returneaza polinomul nul.

PolController

Aceasta clasa este folosita pentru alipirea patii algoritmice de cea vizuala, ea are doua obiecte: view de clasa PolView si model pentru PolModel. Constructorul clase primeste aceste doua obiecte ca parametri, modifica valorile lor si adauga prin metodele de atasare din view un obiect nou de clasa implementatoare de ActionListener. PolController are trei clase pe care le inglobalizeaza, fiecare dintre ele avand metoda pe actionPerformed pentru indeplinirea functilor asociate. CalcListener este clasa asociata actiuniilor pe care butonul calc trebuie sa le faca, in cazul in care polinoamele sunt introduse gresit se va afisa ”Valori ilegale”. Se extrage starea curenta a ferestrei si in functie de valoarea ei se va merge pe o ramura a switch-ului:

- ‘1’-adunare. Se extrag sirurile si prin metoda addString se formeaza doua polinoame. Suma lor se face in alt polinom folosind metoda “add”.

Rezulatatul este setat Jlabel-ului aferent starii 0: /* view.setPolResult(polAddRes.afisare(), 0); */;

- ‘2’-scadere. Se extrag polinoamele, se face scadere folosind “sub” si se afiseaza rezultatul in JLabel-ul ‘1’;
- ‘3’-derivare Se extrag polinoamele, se face derivarea folosind “derivate” si se afiseaza rezultatul in JLabel-ul ‘2;
- ‘4’-integrare. Se extrag polinoamele si se integreaza polinomul introdus prin functia “integrate”. Rezultatul este afisat;
- ‘5’-inmultire. Se inmultesc prin metoda “mul” datele de intrare si se afiseaza rezultatul;
- ‘6’-impartire. Se preiau datele de intrare si cu ajutorul functilor “iMulQ” si “iMulR” se determina catul si restul care se afiseaza.

Clasa CalcListener prin metoda actionPerformed seteaza state la 7, numar ce corespunde ferestrei de meniu. ChooseListener contine in metoda actionPerformed un switch pentru starea selectata a combo-ului choose. In functie de operatia aleasa se seteaza starea conform distributiei de mai sus.

PolTest

PolTest este clasa descrisa pentru testarea operatilor pe polinoame. Ea extinde clasa TestCase si are 6 metode denumite test(Operatie). In fiecare metoda sunt create polinoame, polinoamelor le sunt date valori si se foloseste asertiunea assertTrue sau assertEquals pentru verificare egalitatii dintre polinoamele date si cele rezultate.

Observatii

Datele de intrare pentru polinoame trebuie sa specifice mereu coeficientul si gradul monomului, fiind tinute intr-un ArrayList gradul lor poate sa fie teoretic oricat de mare. Ele sunt de variabila x si trebuie introduse cu atentie deoarece spatile sau aparitia multipla a operatorilor nu sunt agreate, ordinea monoamelor in polinom nu conteaza. Operatile de imultire si impartire ocupa un timp T destul

de crescut, iar pentru aflarea restului la impartire se folosesc trei operatii aditionale care dezavantajeaza numarul de operatii efectuate.

Links

- https://regex101.com/
- http://stackoverflow.com/

Singeorzan Ionut-Florin
Gr: 30227