DOCUMENTATIE

TEMA *1*

NUME STUDENT: Alixandrescu Octavian

GRUPA:30223

# CUPRINS

1. Obiectivul temei 3

2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3

3. Proiectare 6

4. Implementare 7

5. Rezultate 17

6. Concluzii 17

7. Bibliografie 17

# Obiectivul temei

Obiectivul principal: Proiectarea si implementarea unui calculator polinomial cu o interfață grafică dedicată prin intermediul căreia utilizatorul poate insera polinoame, poate selecta ce operații matematice să se efectueze și poate vizualiza rezultatele.

Obiective secundare:

* Analiza problemei si identificarea cerințelor - cap. 2
* Proiectarea calculatorului polinomial - cap. 3
* Implementarea calculatorului polinomial - cap. 4
* Testarea calculatorului polinomial - cap. 5

# Analiza problemei, modelare, scenarii, cazuri de utilizare

Cerinte functionale:

* Calculatorul polinomial ar trebui să permită utilizatorului să insereze polinoame
* Calculatorul polinomial ar trebui să permită utilizatorului să selecteze operația matematică
* Calculatorul polinomial ar trebui să adune două polinoame
* Calculatorul polinomial ar trebui să scadă două polinoame
* Calculatorul polinomial ar trebui să înmulțească două polinoame
* Calculatorul polinomial ar trebui să afișeze rezultatul
* Calculatorul polinomial ar trebui să recunoască dacă datele de intrare nu sunt corespunzătoare și să ceară utilizatorului noi date de intrare care să descrie corect un polinom

Cerinte non-functionale:

* Calculatorul polinomial ar trebui să fie intuitiv și ușor de folosit de către utilizator
* Calculatorul polinomial ar trebui să fie implementat folosind paradigma programării orientate pe obiect
* Calculatorul polinomial ar trebui să fie implementat cu algoritmi eficienți din punctul de vedere al timpului de execuție și al memoriei folosite

**Use case:** adunarea polinoamelor

**Primary actor:** utlizatorul

**Sccenariul principal terminat cu succes:**

1. Utilizatorul inserează două polinoame în interfața grafică

2. Utilizatorul selectează operația de adunare

3. Utilizatorul apasă butonul de calculare

4. Calculatorul polinomial execută operația de adunare a celor două polinoame introduse și afișează polinomul rezultat

**Secvența alternativă:**

Polinoame incorecte - forma polinoamelor introduse de către utilizator este incorectă

-se reia scenariul de la pasul 1.

**Use case:** scăderea polinoamelor

**Primary actor:** utlizatorul

**Sccenariul principal terminat cu succes:**

1. Utilizatorul inserează două polinoame în interfața grafică

2. Utilizatorul selectează operația de scădere

3. Utilizatorul apasă butonul de calculare

4. Calculatorul polinomial execută operația de scădere a celor două polinoame introduse și afișează polinomul rezultat

**Secvența alternativă:**

Polinoame incorecte - forma polinoamelor introduse de către utilizator este incorectă

-se reia scenariul de la pasul 1.

**Use case:** înmulțirea polinoamelor

**Primary actor:** utlizatorul

**Sccenariul principal terminat cu succes:**

1. Utilizatorul inserează două polinoame în interfața grafică

2. Utilizatorul selectează operația de înmulțire

3. Utilizatorul apasă butonul de calculare

4. Calculatorul polinomial execută operația de înmulțire a celor două polinoame introduse și afișează polinomul rezultat

**Secvența alternativă:**

Polinoame incorecte - forma polinoamelor introduse de către utilizator este incorectă

-se reia scenariul de la pasul 1.

**Use case:** derivarea polinoamelor

**Primary actor:** utlizatorul

**Sccenariul principal terminat cu succes:**

1. Utilizatorul inserează un polinom în interfața grafică

2. Utilizatorul selectează operația de derivare

3. Utilizatorul apasă butonul de calculare

4. Calculatorul polinomial execută operația de derivare a polinomului introdus și afișează polinomul rezultat

**Secvența alternativă:**

Polinom incorecte- forma polinomului introdus de către utilizator este incorectă

-se reia scenariul de la pasul 1.

# 3. Proiectare

Structura calculatorului polinomial este una simplă. Dacă ar fi să privim calculatorul polinomial ca pe o entitate, inputurile lui ar fi cele două polinoame si operația matematică ce trebuie aplicată, iar outputul ar fi polinomul rezultat.

Graphical user interface

Description automatically generated

Aceasta este diagrama UML a calculatorului polinomial. Se observă ca proiectul conține 4 clase, inclusiv clasa Main. Clasa Monomial contine 2 fielduri, putere si coeficient. Clasa Polynomial contine o listă de obiecte de tip Monomial. Clasa GUI este clasa răspunzătoare de crearea interfeței grafice prin care să comunice cu utilizatorul. Practic, clasa GUI este un intermediar între utilizator și clasele Monomial si Polynomial. Ca strcturi de date se poate observa că se folosește *ArrayList* și structuri specifice Java Swing , în cadrul clasei GUI. Nu au fost definite interfețe specifice pentru acest proiect, nici nu au fost folosiți algoritmi speciali. Pentru a extrage puterea si coeficientul fiecărui monom dintr-un polinom, am folosit regular expression and pattern matching, în aceeași manieră în care este descris în prezentarea suport a temei.

# 4. Implementare

**4.1** Clasa Monomial

Graphical user interface, text, application, chat or text message

Description automatically generated

Clasa Monomial este practic clasa de baza a calculatorului polinomial. Ea contine 2 variabile, putere si coef, suficiente pentru a descrie un monom. De asemenea, ea conține și metode reprezentând operațiile de bază: adunare, scădere, înmulțire și derivare. Pe lângă acestea, regăsim si settere si gettere pentru cele două fielduri amintite mai sus, folosirea acestora fiind necesară pentru a implementa operațiile de bază.

La nivelul aceste clase, se află și metoda valid(), care validează datele transmise de către utilizator.

Text

Description automatically generated

Clasa de mai sus face mai multe verificări asupra stringului formatat, înainte de a crea un obiect de tip Monomial. În primul rând verifică dacă stringul dat conține caractere speciale (?, !, @ etc) care nu au ce căuta în textul unui polinom. Apoi se verifică dacă textul conține caracterul ^ și se tratează situațiile cand acesta există și cand nu există. În final, se caută coeficienții din fața lui x, si se tratează cazul cand nu există nimic în fața acestuia, sau cazul în care nu există x. Această metodă este apelată dintr-o alta din clasa Polynomial, despre care se va discuta în cadrul acelei clase.

De asemenea clasa Monomial conține și metoda comparatorPutere, folosită în cadrul înmulțirii polinoamelor pentru a sorta polinomul descrescător în functie de gradul monoamelor, pentru că, în urma algoritmului din metoda înmulțire din cadrul clasei Polynomial, lista de monoame ar putea fi nesortată la final și necesită o sortare pentru a afișa rezultatul după forma standard.

Text

Description automatically generated

**4.2** Clasa Polynomial

Graphical user interface, text, application

Description automatically generated

Clasa Polynomial este clasa care permite crearea polinoamelor, ca și colecții de monoame. La nivelul ei se regăsește un constructor public, care necesită un *ArrayList* de obiecte de tipul Monomial și care va crea un nou obiect de tipul Polynomial pe baza ArrayList-ului inserat.

* Adunarea a două polinoame

La prima vedere, adunarea polinoamelor se rezumă la a însuma coeficienții monoamelor care au gradul egal, însă implementarea doar a acestei idei nu este valabilă pe un caz general.

Text

Description automatically generated

Algoritmul de mai sus împarte adunarea pe mai multe cazuri. În primul rând, daca se epuizează monoamele într-unul dintre cele două polinoame, dar încă există monoame în celălalt polinom, pur și simplu se adaugă monoamele rămase la rezultat. În cazul în care există monoame în ambele polinoame, trebuie să tratăm cazurile în funcție de gradele ambelor monoame la care ne află, în moemntul actual. Dacă gradele sunt egale, atunci se realizează operația de adunare a celor două monoame și se adauga listei monomul rezultat. Pentru cazul în care gradele nu sunt egale, se adaugă în listă monomul cu grad mai mare și se referențiază doar iteratorul respectiv.

* Scăderea a două polinoame

Algoritmul de scădere a două polinoame este identic cu cel de adunare din perspectiva parcurgerii celor două polinoame. Pentru cazul în care gradele monoamelor sunt egale, se realizează operația de scădere a celor 2 monoame, iar o altă diferență apare în cazul în care gradul celui de-al doilea monom este mai mare decât gradul primului monom, caz în care coeficientul celui de-al doilea polinom va fi înmulțit cu -1 și abia apoi adăugat la polinomul rezultat.

* Înmulțirea a două polinoame

În cazul în care se dorește înmulțirea a două polinoame, fiecare monom din primul polinom va fi înmulțit cu fiecare monom din al doilea polinom. Trebuie verificate cazurile în care, din înmulțirea monoamelor, vor rezulta monoame de același grad, care trebuie adunate și sortate pentru a obține polinoame de formă standard.

Text

Description automatically generated

* Derivarea unui polinom

Algoritmul de derivare a unui polinom este, probabil, cel mai simplu de până acum. Pentru fiecare monom al polinomului se realizează operația de derivare, iar rezultatul se adaugă listei de monoame a polinomului rezultat. Metoda de derivare din cadrul clasei Monomial nu face nimic altceva decât să scadă gradul polinomului cu 1 și sa calculeze noul coeficient ca fiind coeficientul înmulțit cu puterea înainte de modificare.

Text

Description automatically generated

* Conversia din polinom în string

Clasa Polynomial mai conține de asemenea și metoda numită afișare, care, pe baza unui polinom, va returna un string semnificând scrierea polinomului respectiv. Cu ajutorul acestei clase se afișează rezultatul în cadrul interfeței grafice.

Text

Description automatically generated

Se observă că se tratează separat cazurile în care coeficientul unui monom este 1 sau -1, pentru că in acest caz nu se va afișă coeficientul ca în restul cazurilor ci se va afișa nimic sau – în fața lui x. De asemenea, pentru fiecare dintre cele 4 cazuri se tratează și cazurile speciale în care puterea este 0 sau 1, conversia făcându-se în mod diferit pentru aceste cazuri. La final se verifică dacă există caracterul + la începutul polinomului, și se șterge caracterul în caz afirmativ.

* Conversia din string în polinom

Metoda care este responsabilă de conversia unui string într-un obiect al clasei Polynomial este metoda stringToPoli în cadrul căreia se împarte stringul dat într-un array de stringuri, folosind ca delimitator caracterul +. Apoi, pentru fiecare string din cadrul array-ului se execută metoda valid, din clasa Monomial. În fapt, munca laborioasă de parsare a stringului primit ca input este practic făcută în metoda din clasa Monomial, despre care s-a discutat mai sus. Cazurile în care metoda stringToPoli returnează valoarea false sunt cazurile în care utilizatorul nu a introdus un polinom în forma corectă urmând ca utilizatorului să îi fie cerute noi date de intrare.

Text

Description automatically generated

**4.3** Clasa GUI

Graphical user interface, text

Description automatically generated with medium confidence

Clasa GUI este clasa răspunzătoare de crearea interfeței grafice prin intermediul căreia utilizatorul interacționează cu programul. Această clasă extinde clasa JFrame si implementează interfața ActionListener. Ea conține mai multe variabile de tipul JLabel, JButton, JTextField sau JComboBox, care sunt inițializate în cadrul constructorului. Metoda actionPerformed provine din cadrul interfeței implementate. În cadrul acestei metode se apelează metode din clasa Polynomial, pe baza datelor de intrare furnizate de către utilizator. În cadrul GUI se regăsește și metoda dateIncorecte, care va afișa un mesaj utilizatorului, cerându-i să introducă noi date, și va reseta campurile în care utilizatorul introduce datele. Metoda privată getEmptyLabel este apelată în cadrul metodei dateIncorecte, pentru a facilita crearea noului frame în care se afișează mesajul pentru utilizator. Se observă in codul de mai jos că, în cadrul metodei actionPerformed se verifică operația matematică aleasă de către utilizator, iar pe baza acesteia se apelează metoda potrivită din cadrul clasei Polynomial, pe cele două polinoame primite de la utilizator.

A screenshot of a computer

Description automatically generated with medium confidence

**4.4** Clasa Main

În cadrul clasei Main, se creează un obiect de tipul GUI și se apelează constructorul clasei, pentru a inițializa interfața grafică.

# Rezultate

Nu am testat rezultatele cu ajutorul unui framework de testare. Testarea am realizat-o introducând polinoame care să acopere eventualele cazuri speciale care ar putea apărea așa cum am descris în cadrul descrierii algoritmilor și am comparat rezultatele cu unele obținute manual sau cu ajutorul unor calculatoare polinomial găsite online.

# Concluzii

În această primă temă aș putea spune că cel mai interesant dar și dificil de implementat a fost conversia din String într-un obiect de tip Polynomial, pentru ca există multe cazuri care trebuie tratate pentru a realiza conversia corespunzător, în proporție de 100%. Pot spune ca documentul suport m-a ajutat, prin prezentarea claselor Matcher si Pattern. Cunoașterea acestor clase și a anumitor metode mi-a facilitat proiectarea algoritmului prin care, dintr-un String reprezentând un polinom, creez un obiect al clasei Polynomial, corespunzător polinomului reprezentat prin String. Eventualele dezvoltări ar putea fi, implementarea operației de împărțire a două polinoame, implementarea operației de integrare sau testarea folosind un framework de testare. În urma rezolvării acestei prime teme am învățat lucuri noi legate de limbajul de programare Java, cum ar fi clasele Matcher și Pattern, sau folosirea Iteratorilor, dar mi-am și solidificat cunoștințele anterioare de programare orientată pe obiect, acest proiect fiind unul dintre cele mai ample pe care l-am realizat până acum, folosind această paradigmă și acest limbaj de programare.

# Bibliografie

<https://dsrl.eu/courses/pt/materials/A1_Support_Presentation.pdf>

<https://docs.oracle.com/javase/7/docs/api/java/util/regex/Matcher.html>

<https://docs.oracle.com/en/java/javase/13/docs/api/java.base/java/util/regex/Pattern.html>

<https://docs.oracle.com/javase/8/docs/api/java/util/Iterator.html>