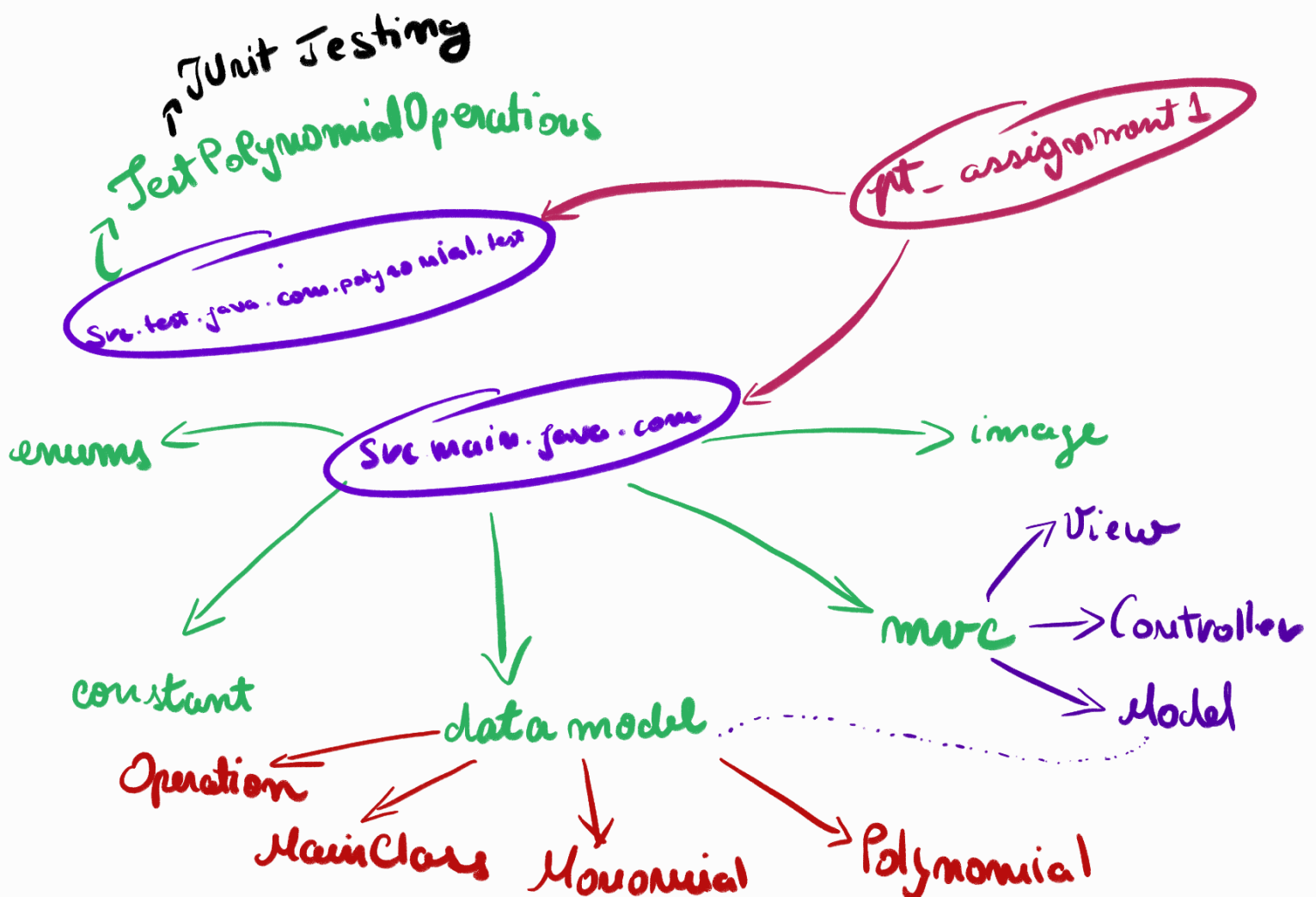


# Polynomial Calculator

Requirement	Grading
<ul style="list-style-type: none"> <li>Use an object-oriented programming design (use encapsulation, define appropriate classes e.g., <i>Polynomial</i> and optionally <i>Monomial</i>).</li> <li>Use Map for modelling the Polynomial. → Hash Map → Polynomial: 7</li> <li>Use <i>foreach</i> instead of <i>for(int i=0...)</i> Polynomial: 51 (entry), 35 (normal)</li> <li>Implement a graphical user interface using Java Swing or JavaFX → mvc → view</li> <li>Implement the addition and subtraction operations Polynomial: 19, 34</li> <li>Implement classes with maximum 300 lines (except the UI classes) and methods with maximum 30 lines.</li> <li>Use the Java naming conventions (see <a href="#">link</a>). package: com.datamodel... etc.</li> <li>Good quality documentation addressing all sections from the documentation template.</li> </ul>	5 p
Good organization of the source code	1 p
Implement the multiplication operation 49	0.5 p
Implement the division operation 57	1 p
Implement the derivative operation 98	0.5 p
Implement the integration operation 107	0.5 p
Use regular expressions and pattern matching for extracting the polynomial coefficients Operation: 7	0.5 p
Use Junit for testing src.test.java.com. ....	1 p

→ Test Polynomial Operations



# com. data models

## → Monomial

- degree, coefficient
- addition, subtraction, multiplication, division, derivative, integral,
- to String → afişarea nr. reale în 2 zecimale
- const, getters, setters
- set Rounding Mode
- aici tratez cazurile pentru afişarea 10, coef. 0, 11, coef 1.

## → Polynomial

↪ `HashMap<Integer, Monomial> poly, remainder`

↪ operații elementare ...

↳ efectuează operații între monomiale, utilizând op. definite în clasa Monom

↳ multiplication: 2 forwori ce iau fiecare element în parte

↳ division: → determină coef. maxim corespunzător celor 2 polinoame

→ A = polinomul "împărțitor" / coef. max

→ B = polinomul Deîmpărțit - A

→ polinomul Deîmpărțit e înlocuit de B

repeat

↪ `toString(String s)` → afişarea restul sau polinomul propriu zis

## → Operation

↳ clasă în care se află metoda `stringWithRegex` prin care se realizează parsarea sirului în elem. Polynomial

# • Con. MVC

◦ **View** → `initElemental()` | → `Listeners ....`  
→ `set Color()`  
→ `set Icon()`  
→ `addElementstoPanels()`  
→ `setLayoutforPanels()`  
→ `set Textformat()`  
.....

## ◦ **Controller**

↳ **metode** → `operation` → în fct. de `Operation Type`, este apelată metoda corrisp. `operation` din `dataModel`

→ `ShowMessageDialog` ⇒ afișarea erorilor într-o nouă fereastră

↳ **class interne** → pt implementarea metodei `actionPerformed`