CS 2336
Project #2 – Derivative Calculator

Summary

This project will require you to create a program that will take a mathematical expression in a single variable (x) and either

- evaluate the expression for a given value of x, or
- calculate the derivative (with respect to x) of the expression

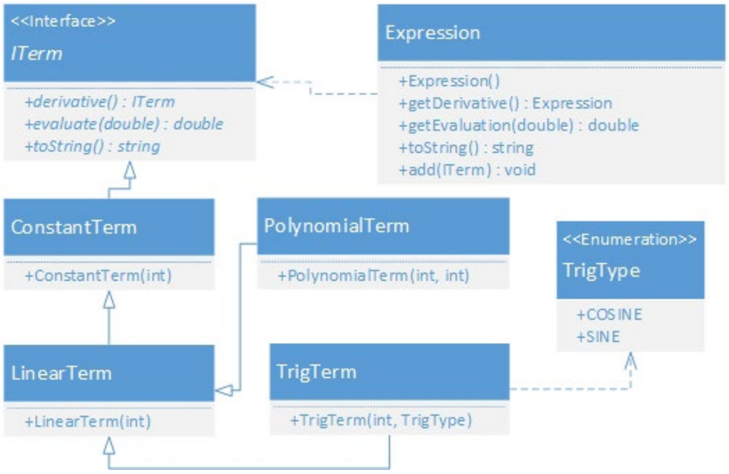
The purpose of this project is to demonstrate a working knowledge of

- Class creation
- Inheritance
- Polymorphism
- Linked Lists
- Templates

This project is designed to be a take-home part of exam #2. You are not to discuss or share code with anyone, including CSMC or tutors. You may use the book, but no other outside materials.

Process

You will create each of the following classes, interfaces, and enumerations according to the UML diagram below. The diagram only shows the public methods necessary for objects to function. You may add any additional private/protected members you deem necessary.



You will also need to create a linked-list template.

- A `ConstantTerm` object will represent a term of the form
  $$\pm a$$
  where a is of type `int`. The derivative of a constant term is always
  $$+0$$

- A `LinearTerm` object will represent a term of the form
  $$\pm ax$$
  where *a* is of type `int` and *x* is the independent variable. The derivative of a linear term is a constant term of the form
  $$\pm a$$

- A `PolynomialTerm` object will represent a term of the form
  $$\pm ax^b$$
  where *a* is of type `int`, *b* is a positive `int` greater than one, and *x* is the independent variable. If *b*>2, the derivative of the polynomial term is a polynomial term of the form
  $$\pm(ab)x^{b-1}$$
  If *b*=2, the derivative of the polynomial term is a linear term of the form
  $$\pm 2ax$$

- A `TrigTerm` object will represent a term of the form
  $$\pm \text{a} \cos{(x)}$$
  or
  $$\pm \text{a} \sin{(x)}$$
  where *a* is of type `int` and *x* is the independent variable. The derivative of the sine term is a trigonometric term of the form
  $$\pm \text{a} \cos{(x)}$$
  The derivative of the cosine term is a sine term of the form
  $$\mp \text{a} \sin{(x)}$$
  Note that the sign flips when taking the derivative of cos(). Evaluation of trigonometric functions should be done in degrees.

- For each subclass of `ITerm`, the `derivative()`, `evaluate()`, and `toString()` functions will need to be overridden.
  - The `getDerivative()` function will create a new `Expression` object containing the derivative of each term of the original object. Zero-valued constant terms should not be included.
  - The `evaluate(double)` will return the sum of the evaluations of the individual terms.
  - The `toString()` function will convert the expression to a string for ease of display. The polynomial terms should be displayed in descending exponential order, followed by linear, constant, sine, and cosine.
  - An `Expression` object will contain a linked list of `ITerm` objects.
  - The list of terms will initially be empty and can be added to via the `Expression` object's `add(ITerm)` method.

`main()`

The `main()` function should create an empty expression object.

Example:

```
AbstractTerm t1 = new LinearTerm(5);
AbstractTerm t2 = new PolynomialTerm(-4,3);
AbstractTerm t3 = new TrigTerm(-6,COSINE);

System.out.println(t1);                 // + 5x
System.out.println(t1.evaluate(5));     // 25

System.out.println(t2);                 // - 4x^3
System.out.println(t2.evaluate(2));     // -32

System.out.println(t3);                 // - 6cos(x)
System.out.println(t3.evaluate(45));    // -4.24

Expression e1 = new Expression();
e1.add(t1);
e1.add(t2);
e1.add(t3);
Expression e2 = e1.getDerivative();
System.out.println(e1);                 // - 4x^3 + 5x - 6cos(x)
System.out.println(e2);                 // - 12x^2 + 5 + 6sin(x)
System.out.println(e1.getEvaluation(0));// -6
System.out.println(e2.getEvaluation(0));// 5
```