

Assignment 3 Story

We were provided MapPoly, an immutable map-based version of Liskov's poly class, and our task was to convert this class into a mutable class.

Most of our code is the same as the code provided to us, we have only changed the methods of *sub()*, *add()*, and *multiply()*.

A walkthrough of our code:

The constructor with no arguments that initializes a zero polynomial has been kept intact. With this, there was a second constructor with two integer parameters that initialized the polynomial $c \cdot x^n$, and throws `IllegalArgumentException` if $n < 0$ has also not been touched.

The method `degree()` returns the degree of the specified polynomial. Another method called `coeff(int d)` returns the coefficient of the term whose exponent is d . Here, if the value of $d < 0$, it throws `IllegalArgumentException`, and if the terms in the polynomial do not contain the exponent ' d ', it returns 0. These two functions haven't been tampered with.

The following methods have been modified to convert this class from immutable to mutable. All of these methods throw `NullPointerException` if the argument object is NULL.

In the method `sub()`, we added a local copy of the class object to prevent modifying original object. We then copied all the terms of original object to this copy, and modified the call to the `add` method by changing its argument from original object to this copy of the object. This call invokes the '`.minus()`' method which also needed modification.

We decided to get rid of the original code, and just replaced all the term values with negative values.

The method `add()` has been modified by first removing a new object initialization. Now we're working with the same class object and hence we'll return it (`return this`), instead of returning another class object in the case of immutable version (`return result`).

Similarly, in the method `mul()`, we removed the new object initialization, and added initialization of a `TreeMap` object with integers. We then modified the rest of the code in this method to populate the `TreeObject` with the product coefficients.