Credit where credit is due:

These problems are adapted from Ilkka Kokkarinen's excellent "CCPS 209 Computer Science II Labs" problem set. Credit for these problems goes to him, and the full set of 99 can be found at his Github page:

https://github.com/ikokkari

Polynomial II: Arithmetic

This lab continues with the Polynomial class from the previous lab by adding new methods for polynomial arithmetic to its source code. No inheritance or polymorphism is yet taking place in this lab. Since the Polynomial type is intentionally designed to be immutable, none of the following methods should modify the objects this or other in any way, but always return the result of that arithmetic operation as a new Polynomial object created inside that method.

`public Polynomial add(Polynomial other)`

Creates and returns a new Polynomial object that represents the result of polynomial addition of the two polynomials this and other. Make sure that the coefficient of the highest term of the result is nonzero, so that adding two polynomials $5x^{10} - x^2 + 3x$ and $-5x^{10} + 7$, both having the same degree of 10, produces the result $-x^2 + 3x + 7$ that has a degree of only 2. As the JUnit test class generates random polynomials and adds them together, occasionally some polynomial will be added to its own negation, producing the zero polynomial as the result. Your code must handle this important edge case correctly.

`public Polynomial multiply(Polynomial other)`

Creates and returns a brand-new Polynomial object that represents the result of polynomial multiplication of this and other. You can perform this multiplication by looping through all possible pairs of terms between the two polynomials and adding their products together into the result where you combine the terms of equal degree together into a single term. Multiplication of two polynomials can cancel out some internal terms whose coefficients were originally nonzero in both polynomials. For example, multiplying $x^2 + 3$ by $x^2 - 3$ gives the result $x^4 - 9$ whose second order term ends up with a zero coefficient and vanishes from the result. However, unlike when adding two polynomials, the highest term can never get cancelled out this way, since the product of two nonzero integer coefficients can never be zero. You therefore know the degree of the entire result of multiplication right away once you know the degrees of the original polynomials, without actually having the carry out this multiplication by adding all the terms together.