Jordan Chin
Prof. Rui
CMPU 102

Report on Polynomial Calculator

As the main structure for my Polynomial Class, I used a TreeMap<Integer,Float>, where the exponent of each element of a Polynomial would be the integer and the coefficient would be the float. The TreeMap was chosen due to it being sorted, which contributed to ensuring the String representation was always in order of highest to lowest exponents, and all keys being unique, so I can be assured that no more than one exponent would be represented in the set at any one time. Also, using only a TreeMap and building on it made my code easier and reduced the layers of going within methods to extract useful information.

In order to go through arithmetic expressions effectively, I decided to use a finite state machine, taking into account the different forms monomials could take. In addition, I separated the process of parsing, converting to postfix, and evaluating into three separate classes in order to make the process less stressful and more organized. The WSIG state was more of a universal state, being the filter for most of the other states. The WINT and INT states both served in getting the coefficient, but since constants can exist, it also took that into consideration. A special provision had to be made for the case where a positive or negative sign could be in front of the string or next to an open bracket, where no operation could occur even though it is implicit. Thus, I had to insert a zero Polynomial so the operation could occur without any crashes occurring.

As for special features, my calculator was made using only GUI, and I created a TextArea for the Files, so that everything would automatically run when the file is chosen. Unfortunately for limitations, I had not fully differentiated between the different errors that could arise during the running of the program, so the same error message would be showing for different scenarios. And as for code, some classes are more diligently commented on than others.

In conclusion, the Polynomial calculator employes some of Java's utilities which make running the calculator much easier than it would have been otherwise. Despite its shortcomings, I believe that it is makes for it with its TreeMap implementation and FSM utilization.