INSTITUTO TECNOLOGICO DE COSTA RICA **SEMESTRE 1, 2016**

INGENIERIA EN COMPUTACIÓN

LENGUAJES DE PROGRAMACIÓN – IC4700

DOCUMENTACIÓN TAREA PROGRAMADA – EIFFEL

CALCULADORA DE POLINOMIOS DE DOS VARIABLES

1. Introduccion

   "La tarea consiste en programar mediante en Eiffel las clases necesarias para implementar operaciones con polinomios de dos variables. La funcionalidad de los polinomios deberá encapsularse de tal manera que sea posible usar dicha clase para implementar un ciclo que realiza operaciones de una calculadora tipo notación polaca inversa."

   Basicamente eso es en lo que consiste el proyecto, crear un programa el cual pueda realizar operaciones de suma, resta y multiplicacion (ademas de otras funciones, que se listan mas adelante) utilizando el metodo polaco inverso.

   Se debe de tomar en cuenta factores de elementos negativos, mostrar elementos segun su indice, y simplificar la expresion lo mas posible. A continuacion se muestra informacion relacionada al desarrollo y fin del proyecto.

2. Ambiente de desarrollo

   El proyecto fue desarrollado en una maquina con windows 8.1(version de 64 bits) sistema operativo bajo el cual se utilizo la herramienta de desarrollo EiffelStudio 15.12.9.8497 Enterprise Edition – win65 y con ello los compiladores que trae consigo esta herramienta.

3. Estructuras de clases usadas

   Tripleta: Clase la cual contiene los valores de multiplicador y sus respectivas variables(X,Y). Cuenta con metodos set's y get's lo cuales permiten acceder a los datos ademas de otros metodos como "evaluar" el cual a como lo indica su nombre permite evaluar la tripleta dada segun un valor a sus variables. Basicamente contiene informacion sobre las tripletas y provee metodos auxiliares para el desarrollo del proyecto.

   Polinomio: Clase la cual contiene un conjunto de tripletas modelando asi un "polinomio", sobre el cual se aplican operaciones, un ejemplo de metodo es el de simplificar(dado un polinomio este se simplifica asi mismo), basicamente es la estructura de datos principal del programa.

   Parser: Clase la cual provee un conjunto de operaciones para el tratamiento de cadenas de texto, a como lo dice el nombre, parsea o divide la informacion contenida en una linea. Se creo esta clase debido a que se quiere mantener el principio de simplicidad en el programa, mostrandolo de la forma mas modular posible.

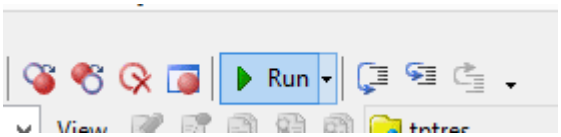
Main: Clase la cual controla el flujo principal del programa, esta es la encargada de solicitar datos al usuario y asi poder transmitir informacion a otras clases. Basicamente en esta clase se encuentra el menu, es la encargada del controlar el flujo principal del programa y se encarga de controlar la entrada de los datos.

Operacion: Clase la cual cuenta con atributos y metodos necesarios para la manipulacion de las estructuras de datos y asi poder realizar operaciones de suma, resta y multiplicacion con los polinomios.

Relacion entre las clases: En general la relacion que entre las clases en al ambito de estructura de datos es que un POLINOMIO esta conformado de TRIPLETAS, MAIN es la encargada de obtener las instrucciones provenientes del usuario y parser interpreta las mismas, tambien realizar llamadas a polinomios y realiza operaciones gracias a la clase OPERACION. Ademas de las clases mencionadas en este apartado, se utilizo la clase ARRAYED_STACK para el manejo de la pila.

4. Instrucciones para ejecutar el programa

Para poder ejecutar el programa es necesario contar con un IDE que ya cuenta con cada uno de los elementos necesarios para poder ejecutar el programa. Si se cuenta con este, basicamente basta abrir el proyecto que sigue la estructura presentada en un punto siguiente y presionar el boton de “Correr” o “Run” presente en el programa.

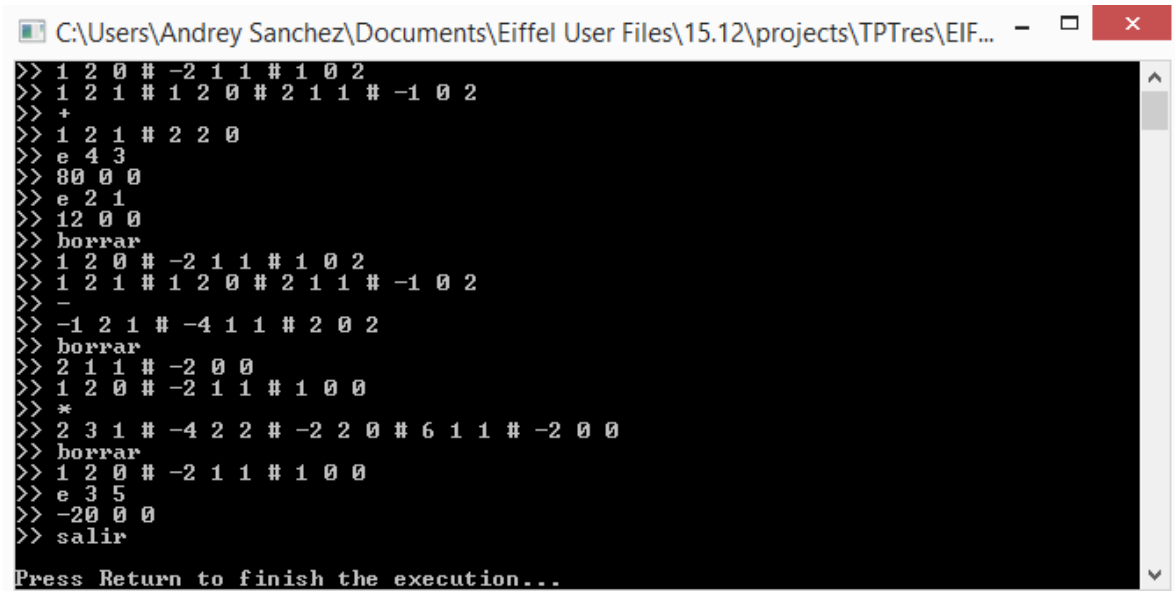


Una vez realizado esto es posible ingresa el conjunto de instrucciones dadas en la especificacion del proyecto.

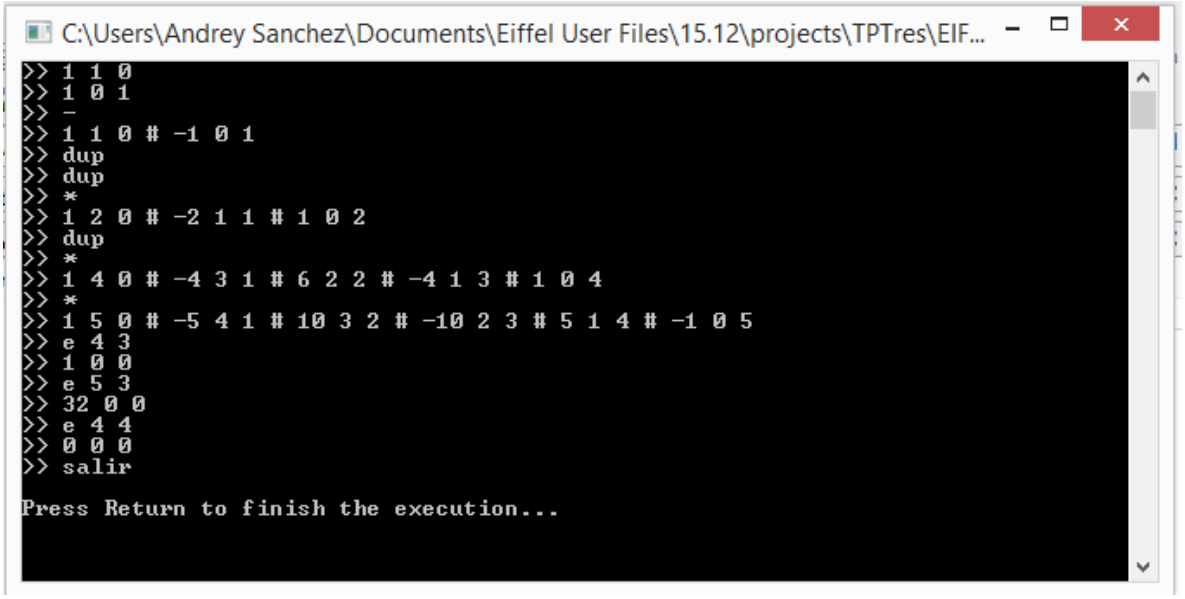
5. Corridas de ejemplo

A continuacion se muestra las corridas ejemplo y su EVIDENCIA de que el programa funciona correctamente en un 100%(Se da a entender que todas las pruebas fueron exitosas, por lo que se omite poner las tablas de las corridas e indicando si fue exitosa o no la prueba)

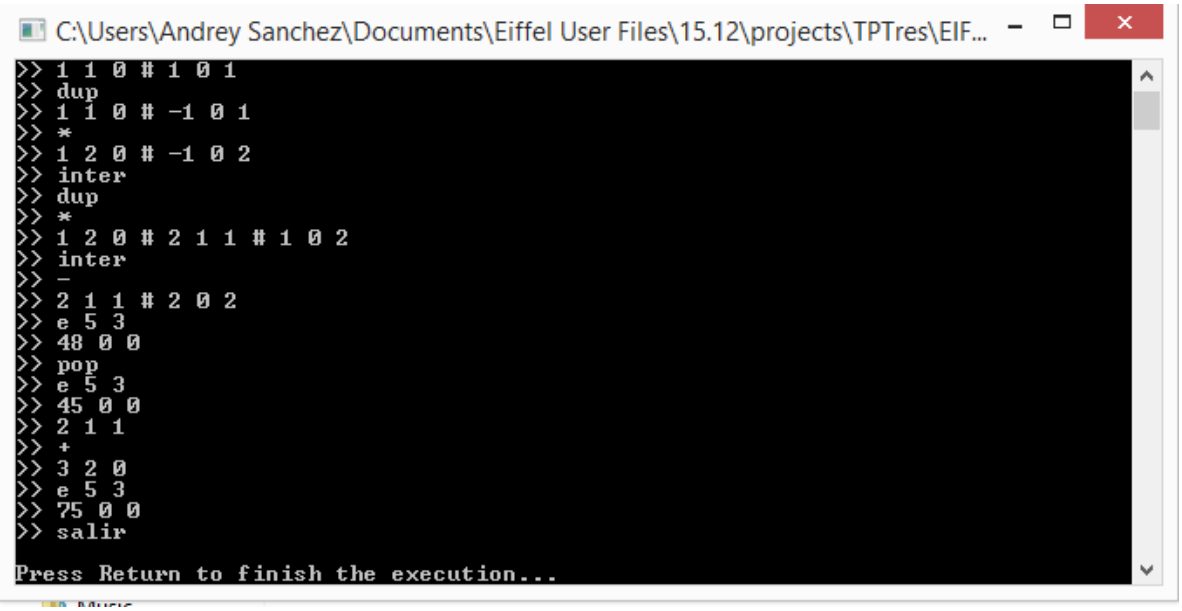
Corrida # 1



Corrida # 2



Corrida # 3



6. Descripcion de algunas features

Basicamente la forma en la que la calculadora evalua polinomios es dada una estructura de datos que contiene tripletas que representan el numero y el exponente de las variables. Dada un conjunto de tripletas se forma un polinomio, para poder realizar sumas o restas se utliza la concatenacion de polinomios añadido un factor multiplicativo o no y despues de esto se procede a simplificar la estructura resultante. Para la evaluacion de los polinomios basta con evaluar cada una de las tripletas con los valores de “x” y “y” dados y realizar la suma de todas las tripletas.

Asi mismo en otro apartado se menciona sobre la toma de entradas por parte del usuario y su correspondiente interpretacion.

Existen features de accesabilidad, de set’s, get’s y operaciones con la estructura de la clase que no hace falta de explicar debido a su trivialidad

Mecanismos de programacion por contrato:

Algunos ejemplos son los siguientes:

```
setP1(T:POLINOMIO)
    require
        T.elements.count>0
    do
        poli1:=T
        ensure
            poli1.elements.count>0
    end
```

Se debe de estar seguro que no se va a realizar una operacion con un polinomio que no tenga elementos para evitar errores en el programa.

```
invariant
    list.count<4
end
```

Se debe de contar con un polinomio de no mas de 2 variables, por lo que con esta condicion se logra facilmente.

Anteriormente se expuso situaciones en las que se utilizo contratos en eiffel, existen mas pero debido a su trivialidad no se mostraran(en caso de que se deseen conocer, revisar el codigo fuente del proyecto)

7. Comentarios finales

El programa es en un 100% funcional, no cuenta con ninguna clase de error o limitacion.