Inclusiones

Declaro las variables necesarias

Mientras Comando distinto de salir

Imprimo “Ingrese comando:”

Guardo lo ingresado en un string

Parseo Cacomando(string, string) //Aquí se valida que exista el comando

Según Comando Hacer

Caso crear: crear(string), Break.

Caso sumar: sumar(string), Break.

Caso multiplicar: multiplicar(string), Break.

Caso evaluar: evaluar(string, int), Break.

Caso esraiz: esraiz(), Break.

Caso mostrar: mostrar(), Break.

Caso guardar: guardar(), Break.

Caso recuperar: recuperar(), Break.

FinSegun

Fin Mientras

sumar(string){

Validarsumar()

Creo un nuevo polinomio vacío con el nombre especificado

Realizo la suma de los otros polinomios

Muestro el resultado

}

evaluar(string, int){

Validarmultiplicar()

Creo un nuevo polinomio vacío con el nombre especificado

Realizo la multiplicacion de los otros polinomios

Muestro el resultado

}

|  |  |
| --- | --- |
| IngresarComando | |
| Entrada: | Caracteres ingresados por teclado |
| Salida: | - |
| Método: | Creo un string nulo (strcrear) Cargo lo ingresado en un string dinámico (scan) Parsear el string ingresado |

|  |  |
| --- | --- |
| Parsear | |
| Entrada: | String de comandos  Lista de String vacía por referencia |
| Salida: | Lista de String cargada |
| Método: | Recorro String  Mientras caracter sea distinto a fin de línea  Si carácter es distinto a espacio  Cargo String con carácter  Sino  Agrego caracter fin de línea  Guardo nuevo String en lista de String  FinMientras |

|  |  |
| --- | --- |
| ValidarComando | |
| Entrada: | Lista de String cargada con el comando y parametros |
| Salida: | TRUE,FALASE |
| Método: | Tomo primer valor de la lista y lo elimino de ella.  EnCasoQueSea  crear: ValidoCrear  sumar: ValidoSumar  multiplicar: ValidoMultiplicar  evaluar: ValudoEvaluar  esraiz: ValidoEsRaiz  mostrar: ValidoMostrar  guardar: ValidoGuardar  recuperar: ValidoRecuperar  salir: abandona la aplicación  OTRO: mensaje de error FinEnCasoQueSea |

|  |  |
| --- | --- |
| ValidoCrear | |
| Entrada: | Lista de String cargada parámetros |
| Salida: | TRUE,FALASE |
| Método: | Tomo primer valor de la lista y lo elimino de ella.  Si es un string  Si su nombre no está en el ABBPolinomio  Creo un nuevo polinomio con dicho nombre  Sino  Mensaje de error  Sino  Mensaje de error  FinSi |

|  |  |
| --- | --- |
| Multiplicar | |
| Entrada: | 2 Listas de Terminos Polinomio pasado por referencia |
| Salida: | - |
| Método: | Mientras que la primer lista no sea nula  Obtento termino lista 1  Mientras que la segunda lista no sea nula  Obtengo termino lista 2  Multiplico termino lista 1 con termino lista 2  Agrego termino al Polinomio por referencia  Sigo recorriendo lista 2  FinMientras  sigo recorriendo lista 1 FinMientras Muestro Polinomio final |

|  |  |
| --- | --- |
| Evaluar | |
| Entrada: | Lista de Terminos Entero para evaluación |
| Salida: | Valor del polinomio dado el entero de entrada |
| Método: | Mientras que la lista no sea nula  Obtengo termino  Multiplico su base por el entero de entrada  Multiplico el resultado por si mismo tantas veces como el exponente FinMientras Devuelvo valor por pantalla |

|  |  |
| --- | --- |
| Crear | |
| Entrada: | String Lista de Terminos Polinomio pasado por referencia |
| Salida: | - |
| Método: | Agrego el string como nombre del Polinomio. Mientras que la lista no sea nula  Agrego el termino al principio de la lista de terminos del polinomio  FinMientras |

MÓDULO: