PRÁCTICA

CURSO	:	***Algorítmica II***
No.	:	*Práctica No.04*
TEMA	:	*Divide y Vencerás – Calculadora*
DURACIÓN ESTIMADA	:	*01:40 horas.*

I. OBJETIVOS

La presente práctica tiene por objetivos:

- Utilizar oLoop para la especificación de pseudocódigo.
- Desarrollar una solución aplicando apuntadores y listas enlazadas.

II. RESUMEN

Debe desarrollar una aplicación en oLoop que se encargue de administrar el diccionario de palabras sinónimas. Considere que debe la clase Nodo que contendrá la información y el apuntador del siguiente elemento de la lista.

III. CONSTRUCCIÓN DE LA APLICACIÓN

```
//primera versión usando reemplazo
Clase Ccalc
   Atributos
        acumulador
        operando
        visor            //este atributo puede ser eliminado es equivalente al acumulador
        operadorAnterior
   Metodos
        constructor() virtual
        destructor
        borra
        borraTodo
        opera(nSigno) virtual
        ingresaValor(nValor)
        MuestraValor→
fClase

Metodo Ccalc.Constructor()
        acumulador ← 0
        operadorAnterior ← 0
        operando ← 0
        visor ← 0
fMetodo

Metodo Ccalc.Borra
        operando←0
        visor = 0                    // versión anterior de  visor←acumulador
fMetodo
```

```
Metodo Ccalc.BorraTodo
        acumulador ← 0
        operadorAnterior ← 0
        operando ← 0
        visor ← 0
fMetodo

Metodo Ccalc.opera(nSigno)
        Si ( operadorAnterior = 0 )   Entonces
                acumulador←operando
                operadorAnterior←nSigno
        Sino
                Caso ( operadorAnterior ) vale
                        1: acumulador←acumulador + operando
                        2: acumulador←acumulador - operando
                        3: acumulador←acumulador * operando
                        4: acumulador←acumulador / operando
                fCaso
                Si ( nSigno = 5 ) Entonces
                        // operadorAnterior ← 0       elimina el operador anterior en la
                                                      // versión anterior
                        operando ← acumulador
                Sino
                        operadorAnterior ← nSigno
                fSi
        fSi
        visor ← acumulador
        operando ← 0
fMetodo

Metodo Ccalc.IngresaValor( nValor )
        // ingresa dígito a dígito
        // nValor es un solo dígito
        // podría ser manejado en la interfaz
        // y este método tener como parámetro el operando
        dígitos ← 0
        num ← operando
        Mientras ( num ≠ 0 ) hacer
           num←num DIV 10
           dígitos ← digitos + 1
        fMientras
        // máximo número de dígitos 10, si tiene más de 10 dígitos no anexa el dígito
        // nValor a la derecha
        Si  ( n < 10 ) Entonces
           Si  ( operando = 0 ) Entonces
                operando←nValor
           Sino
                operando←operando*10+nValor
           fSi
           visor ← operando
        fSi
fMetodo
```

```
Metodo Ccalc.MuestraValor →
        retornar visor
fMetodo

Metodo Ccalc.Retroceder
        operando←operando / 10  //elimina el dígito de la derecha tarea para el alumno
        visor←operando
fMetodo
```

```
Clase CcalcF viene-de Ccalc
   Metodos
        opera (nSigno) sobreescribe
fClase

Metodo Ccalc.operaBasico(nSigno)
        //refina método del padre
        Si (operadorAnterior = 0 )   Entonces
           acumulador←operando
           operadorAnterior←nSigno
        Sino
              Caso ( operadorAnterior ) vale
                    1: acumulador←acumulador + operando
                    2: acumulador←acumulador - operando
                    3: acumulador←acumulador * operando
                    4: acumulador←acumulador / operando
                    6: acumulador← (acumulador * operando) /100
                          // Calculo del porcentaje %
                    7: //calcula potencia
                          nBase ← acumulador
                          nPotencia ← 1
                          Para I  desde  1  a  operando hacer
                                nPot ← nPot * nBase
                          fPara
                          acumulador ← nPot
              fCaso
              Si ( nSigno = 5 ) Entonces
                    // operadorAnterior ← 0          // elimina el operador anterior en la
                    // versión anterior operando←acumulador
              Sino
                     operadorAnterior←nSigno
              fSi
        fSi
        visor ← acumulador
        operando ← 0
fMetodo
```

```
Clase CCalcM viene-de CCalc
    Atributos
                memoria
    Metodos
        Constructor( ) sobreescribe
        guardaMemoria
        sumaMemoria
        muestramemoria →
        borraMemoria
fClase

Metodo CcalcM.Constructor
        heredado-de Constructor
        memoria ← 0
fMetodo

Metodo CcalcM.BorraMemoria
    memoria ← 0
fMetodo

Metodo CcalcM.GuardaMemoria
    memoria ← operando
fMetodo

Metodo CcalcM.SumaMemoria
    memoria ← memoria + operando
fMetodo

Metodo CcalcM. muestraMemoria →
    muestraMmemoria ← memoria
fMetodo
```

```
// La idea es que el usuario de la clase diseñe su propia interfaz y use objetos de la
// Jerarquía usando  calculadora
// Se utiliza la variable calcula que es un objeto de la clase Ccalc

Metodo CAcalculadora.jButtonN1_AlClick
        calcula.ingresaValor(1)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonN2_AlClick
        calcula.ingresaValor(2)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonN3_AlClick
        calcula.ingresaValor(3)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonN4_AlClick
        calcula.ingresaValor(4)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonN5_AlClick
        calcula.ingresaValor(5)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonN6_AlClick
        calcula.ingresaValor(6)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonN7_AlClick
        calcula.ingresaValor(7)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonN8_AlClick
        calcula.ingresaValor(8)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonN9_AlClick
```

```
        calcula.ingresaValor(9)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonN0_AlClick
        calcula.ingresaValor(0)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonC_AlClick
        calcula.borra
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonCe_AlClick
        calcula.borraTodo
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonRetroceder_AlClick
        calcula.retroceder
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonMas_AlClick
        calcula.opera(1)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonMenos_AlClick
        calcula.opera(2)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonPor_AlClick
        calcula.opera(3)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonEntre_AlClick
        calcula.opera(4)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo
```

```
Metodo CAcalculadora.jButtonIgual_AlClick
        calcula.opera(5)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonpPorcentaje_AlClick
        calcula.opera(6)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonPotencia_AlClick
        calcula.opera(7)
                //se debe convertir calcula.MuestraValor en string
        jTextFieldNumeros.setText(calcula.MuestraValor)
fMetodo

Metodo CAcalculadora.jButtonMC_AlClick
        calcula.BorraMemoria()
fMetodo

Metodo CAcalculadora.jButtonMS_AlClick
        calcula.GuardaMemoria()
fMetodo

Metodo CAcalculadora.jButtonMR_AlClick
                //se debe convertir calcula.MuestraMemoria en string
        jTextFieldNumeros.setText(calcula.MuestraMemoria())
fMetodo

Metodo CAcalculadora.jButtonMMas_AlClick
        calcula.SumaMemoria()
fMetodo
```

```
//segunda versión usando más características de POO
Clase Ccalc
Atributos
        acumulador
        operando
        visor           //este atributo puede ser eliminado es equivalente al acumulador
        operadorAnterior
Metodos
        constructor
        destructor
        borra
        borraTodo
        operaBasico (nSigno)
        opera(nSigno) virtual
        ingresaValor(nValor)
        MuestraValor→
fclase

Metodo Ccalc.constructor
        acumulador←0
        operadorAnterior←0
        operando←0
        visor←0
fmetodo

Metodo Ccalc.borra
        operando←0
        visor←acumulador
fmetodo

Metodo Ccalc.borraTodo
        acumulador←0
        operadorAnterior←0
        operando←0
        visor←0
fmetodo

Metodo Ccalc.operaBasico(nSigno)
// metodo protegido que ejecuta las operaciones básicas
Si operadorAnterior =0
        Entonces acumulador←operando
                 operadorAnterior←nSigno
        Sino Caso operadorAnterior vale
                1: acumulador←acumulador+operando
                2: acumulador←acumulador-operando
                3: acumulador←acumulador*operando
                4: acumulador←acumulador Div operando
             Fcaso
fmetodo
```

```
Metodo Ccalc.opera(nSigno)
        OperaBasico
        Si nSigno = 5
        Entonces operadorAnterior←0
                 operando←acumulador
        sino  operadorAnterior←nSigno
        Fsi
        visor←acumulador
        operando←0
fmetodo

Metodo Ccalc.ingresaValor(nValor)
//ingresa dígito a dígito
//nValor es un solo dígito
// podría ser manejado en la interfaz y este método tener como parámetro el operando
        n←0
        num←operando
        Mientras num≠0
        hacer num←num DIV 10
              n←n+1
        fmientras
        // máximo número de dígitos 10, si tiene más de 10 dígitos no anexa el dígito
        // nValor a la derecha
        Si n<10
        Entonces Si operando=0
                    Entonces operando←nValor
                    Sino operando←operando*10+nValor
                 fsi
        visor←operando
        fsi
fmetodo

Metodo Ccalc.MuestraValor→
        Retornar visor
Fmetodo

Metodo Ccalc.retroceder
        operando←operando DIV10
        visor←operando
fmetodo


 Clase CcalcF viene-de Ccalc
Metodos
    operaBasico (nSigno)
    fclase



Metodo Ccalc.operaBasico(nSigno)
//refina método del padre
```

```
heredado-de operaBasico(nSigno)
si operandoAnt > 5
entonces Caso operadorAnterior vale
                6: acumulador← acumulador * (operando /100) // %
                7: //calcula potencia
                   nBase ← acumulador
                   nPot ← 1
                   para nI de 1 a operando
                   hacer nPot ← nPot * nBase
                   fpara
                  acumulador← nPot
                Fcaso
  fsi
fmetodo


Clase CcalcM viene-de Ccalc
Atributos
         memoria
Metodos
        guardaMemoria
        sumaMemoria
        muestramemoria→
        borraMemoria
fclase

metodo CcalcM. borraMemoria
  memoria ← 0
fmetodo


metodo CcalcM. guardaMemoria
  memoria ← operando
fmetodo


metodo CcalcM. sumaMemoria
  memoria ← memoria + operando
fmetodo

metodo CcalcM. muestraMemoria→
 muestraMmemoria ← memoria
fmetodo
```