

Práctica N° 5

Tema: Acciones

Duración: 3 clases

Esta práctica tiene como objetivos

- Abstraer la solución de un problema mediante acciones que expresen solo con nombres las partes del mismo. A continuación resolver estas acciones.
- Introducir el uso de acciones con parámetros y razonar sobre sus ventajas.
- Emplear los distintos tipos de pasaje de parámetros analizando ventajas y desventajas.

Ejercicios propuestos

1) Dada la siguiente acción, especifique estado final de la misma.

Acción SumRes(dato a,b \in Z, dato-resultado c \in Z, resultado multi \in R)

Inicio

a \leftarrow a+1

b \leftarrow b-5

multi \leftarrow a*b

si multi \geq 0

entonces

c \leftarrow c + multi

sino

c \leftarrow c - multi

fsi

Fin

¿Cuál es el estado de los parámetros actuales después de las siguientes invocaciones a la acción SumRes, en los algoritmos que se muestran abajo?

<u>Algoritmo ejemplo1</u> <u>lexico</u> x, y, z, p \in Z <u>Inicio</u> x \leftarrow 5 y \leftarrow 4 z \leftarrow 0 SumRes(x,y,z,p) Escribir (x, y, z, p) <u>Fin</u>	<u>Algoritmo ejemplo2</u> <u>lexico</u> a, b, c, pr \in Z <u>Inicio</u> a \leftarrow 8 b \leftarrow -6 c \leftarrow 10 pr \leftarrow 10 SumRes(b,a,c,pr) Escribir (b, a, c, pr) <u>Fin</u>	<u>Algoritmo ejemplo3</u> <u>lexico</u> x, z, b \in Z <u>Inicio</u> x \leftarrow 0 z \leftarrow 10 b \leftarrow 100 SumRes(1,z,x,b) Escribir (1, z, x, b) <u>Fin</u>
---	--	---

2) Dado el siguiente algoritmo:

Algoritmo AreaFiguras

Lexico

x, y, z, sup \in R

Acción cargar(: a, b \in R; f \in caracter)

Inicio

Escribir('ingrese una t si es un triángulo y una r si es un rectángulo')

Leer(f)

Si f= 'r'

entonces

Escribir('ingrese el 1er lado')

Leer(a)

Escribir('ingrese el 2do lado')

Leer(b)

Sino

Escribir('ingrese la base del triángulo')

Leer(b)

Escribir('ingrese la altura del triángulo')

Leer(a)

fsi

faccion

Acción calcular(: a, b ∈ R; f ∈ carácter; : area ∈ R)

Lexico Local

s ∈ R

Inicio

Si f= 'r'

entonces

area ← a * b

Sino

area ← a * b /2

fsi

faccion

Accion mostrar (: a, b ∈ R; f ∈ carácter; area ∈ R)

Inicio

Si f= 'r'

entonces

Escribir('El área del rectángulo dado por los lados', a, b, 'es: ', area)

Sino

Escribir('El área del triángulo dado por la base y la altura', b, a, 'es: ', area)

fsi

faccion

Inicio {programa principal}

cargar (x,y,z)

calcular(x,y,z,sup)

mostrar (x,y,z,sup)

fin

Determine el tipo de pasaje de parámetros que corresponde dar a las variables pasadas en las acciones cargar, calcular y mostrar.

3) a) Analiza y describe lo que hace el siguiente algoritmo y en base a ello, determina el tipo de pasaje de parámetros de cada una de las acciones que componen al mismo.

b) Desarrolla la acción MostrarMonto que corresponde al algoritmo.

Algoritmo CalcularPagoFormadePago

Léxico

apellidoNombres ∈ Cadena { dato para almacenar el nombre del cliente }

montoCompra ∈ R { dato para almacenar la compra \$ del cliente }

esContado ∈ Lógico { dato que permite identificar la forma de pago }

interesTarjeta ∈ 3..15 { dato para almacenar el interés de la tarjeta }

Acción LeerDatos (apeNombres ∈ Cadena, monto ∈ R, contado ∈ Lógico)

Acción PagoContado (monto ∈ R)

Acción IdentificaTarj(interTarj ∈ 3..15)

Acción PagoTarjeta (monto ∈ R, interTarj ∈ 3..15)

Acción MostrarMonto (apeNombres ∈ Cadena, monto ∈ R)

Inicio {del programa principal}

LeerDatos(apellidoNombres, montoCompra, esContado)

si esContado

entonces {esContado=Verdadero}

PagoContado(montoCompra)

sino {esContado=Falso}

IdentificaTarj(interTarjeta)

PagoTarjeta(montoCompra, interesTarjeta)

fsi

MostrarMonto(apellidoNombres, montoCompra)

Fin.

```

Acción IdentificaTarj(           :interTarj ∈ 3..15)
Léxico local
cod ∈ 1..4 {variable para determinar la tarjeta utilizada}
Inicio

Escribir('Ingrese el número de la tarjeta utilizada -1,2,3 o 4-')
Escribir('1.Master – 2.Visa – 3.Cabal – 4.Cordobesa')
Leer(cod)
segun
  (cod=1):interTarj←10
  (cod=2):interTarj←5
  (cod=3):interTarj←15
  (cod=4):interTarj←3
fsegun
Fin

```

```

Acción PagoContado (           monto ∈ R)
Inicio

si monto>=1000
entonces
  monto←monto-(monto*15/100)
sino
  monto←monto-(monto*10/100)
fsi
Fin

```

```

Acción PagoTarjeta (           monto ∈ R, interTarj ∈ 3..15)
Inicio
  monto←monto+monto*interTarj/100
Fin

```

4) Desarrolla una acción que simule una calculadora. Debe recibir dos números que serán los operandos y un carácter que será el operador. En una variable resultado se almacenará el resultado de aplicar el operador a los operandos. Las operaciones que debe soportar son: '+', '-', '/', '*'. En el caso que se intente la división por cero, la acción escribirá un mensaje 'ERROR' y en la variable resultado almacenará un 99999999.

5) Desarrolle una acción que permita cargar las coordenadas de un punto del plano cartesiano XY.

6) Desarrolle una acción que permita cargar los coeficientes **a** y **b** de una recta cuando es expresada en su forma explícita: $ax + b = y$

7) Desarrollar una acción que lea las coordenadas cartesianas del centro de un círculo y su radio

8) Desarrolle una acción que reciba como parámetros de entrada un punto del plano cartesiano, los coeficientes **a** y **b** de una recta cuando está expresada en la forma explícita ($ax + b = y$) y le asigne a una variable (que será un parámetro de resultado) valor verdadero si el punto pertenece a la recta y sino le asignará valor falso. Reusar las acciones creadas en puntos anteriores que les puedan ser útiles.

9) Desarrolle una acción que reciba como parámetros de entrada un punto del plano cartesiano, las coordenadas del centro de un círculo y su radio, y con esta información le asigne a una variable (que será un parámetro de resultado), valor verdadero si el punto está adentro del círculo y sino le asignará el valor falso. Reusar las acciones creadas en puntos anteriores que les puedan ser útiles.

10) Desarrollar una acción que permita ingresar dos pares de coordenadas cartesianas. Las

mismas representan los vértices opuestos de un rectángulo en el plano cartesiano. Utilizando esos mismos parámetros que se usan para el ingreso de los puntos, los empleará como resultado para almacenar en ellos: en el primer par la esquina inferior izquierda del rectángulo y como segundo par, la esquina superior derecha. Observar que los puntos dados como datos pueden ser de dos cualesquiera de esquinas opuestas en diagonal, del rectángulo. Reusar las acciones creadas en puntos anteriores que les puedan ser útiles.

11) Desarrollar una acción que reciba como parámetros de entrada tres pares de coordenadas cartesianas, se debe asumir que el primer par corresponde a la esquina inferior izquierda de un rectángulo y el segundo par a la esquina superior derecha del mismo rectángulo. El tercer par corresponde a un punto del plano cartesiano. Con esta información la acción debe asignar a una variable que será un parámetro de resultado, el valor verdadero si el punto está adentro del rectángulo y sino le asignará el valor falso. Reutilizar las acciones creadas en puntos anteriores que les puedan ser útiles.

12) Reescriba el algoritmo del ejercicio 19 de la práctica Nro 3 utilizando las acciones desarrolladas en los puntos anteriores.

13) ¿Que hace esta acción?

Acción swap (datos-resultados x,y:enteros)

Inicio

x <---- x + y

y <---- x - y

x <---- x - y

faccion

Nota: que resultado daría esta acción si en un algoritmo la invocamos por ejemplo así

leer (a) { siendo a un número entero }

Swap (a, a)

escribir ('Primer valor', a, 'segundo valor', a)

14) ¿Que hace esta acción?

procedure swap (var x,y: integer);

begin

x := x + y;

y := x - y;

x := x - y;

end;

Nota: que resultado daría este procedure si en un programa la invocamos por ejemplo así

readln (a) { siendo a un número entero }

Swap (a, a)

writeln ('Primer valor', a, 'segundo valor', a)

¿Por qué?

15) Reescribe las acciones (incluso puedes modificar el nombre original por otro más abreviado) que se listan a continuación utilizando parámetros, considera al elegir los mismos que deben permitir a la acción ser reutilizada y presta especial atención a que los cálculos interiores a la acción se realicen sólo con los datos que recibe como parámetros:

a) Acción CalculaSalOrdinario, ver ejercicio 16) de la práctica 3.

b) Acción CalculaSalExtra, ver ejercicio 16) de la práctica 3.

c) Acción CalculaDiaQueSigueMesXXDias, ver ejercicio 21) de la práctica 3.

d) Acción Semáforo, recibe como parámetro un color: rojo, verde o amarillo y debe devolver un parámetro de tipo adecuado, respectivamente: 'Alto', 'Ádelante' o 'Precaución'

e) Acción CondiciónAlumno, recibe tres notas y con ellas la acción calcula la menor de las tres notas y el promedio de las tres notas, con estos resultados informa en un parámetro adecuado la condición del alumno que será: 'Libre' si la menor nota es inferior a 4. 'Regular' si la menor nota es mayor o igual a 4 y menor a 6 y el promedio menor a 7. Por último la condición será

‘Promocionado’ si la menor nota es mayor o igual a 6 y el promedio es igual o mayor a 7.

16) Desarrollar una acción que para un número binario de cuatro cifras, devuelva en una variable resultado su valor en base 10 (decimal). Se estudiarán dos formas del problema según la representación de los datos.

Forma 1: los datos son cuatro enteros (0 o 1). Por ejemplo 1,1,0,1

Forma 2: el dato es un entero de cuatro cifras formado solo por ceros y unos. Por ejemplo 1101.

Plan de clases:

1ra Clase: 1, 2, 3 y 4. Nota: Explicar la diferencia entre los tipos de pasaje de parámetros explicados y el pasaje por referencia de Pascal.

2da Clase: 5, 7, 8 y 11. Nota: para realizar 11 suponga ya definidas las acciones necesarias pedidas en los puntos anteriores, invocarlas con los parámetros adecuados. UTILIZAR tipo compuesto.

3ra Clase: 12, 13, 14, 15.d., 16) en la forma 1.

Hacer en Pascal el ejercicio que indique el profesor de trabajos prácticos y entregarlo en la fecha que se fije con ese objeto.