Sistemas de Tiempo Real.

Memoria de los ejercicios 1, 2, 3, 4 y 5.

Ejercicio 01.-

- Apartado 01 – TIPOS ENTEROS

En este apartado se pide trabajar los tipos de datos en ADA.

En primer lugar se inicializan las variables en las que vamos a almacenar los valores estableciendo el tipo y el rango de valores que podrán tomar.

Una vez hecho esto se les da un valor válido a cada una de ellas y, acto seguido, se realizan las sumas. Estas sumas se almacenan en otras variables creadas para contenerlas.

Finalmente a través de la función “Put()” se muestran por consola.

- Apartado 02 – TIPOS DISCRETOS

En este apartado hemos creado un ‘type’ Semáforo que almacena los valores acotados Rojo, Amarillo y Verde.

Este código inicializa el semáforo a su valor “Rojo” e imprime por pantalla el estado del semáforo.

- Apartado 03 – TIPOS REALES

Se han declarado los ‘types’ A y B respectivamente a los cuales se les ha indicado la precisión que van a soportar.

En el ‘begin’ se les asignan los valores apropiados para la compilación, pues trabajamos en coma flotante, y acto seguido suman y muestran por consola.

Como no son ‘type’ Float, si no A y B, hay que castearlos en la suma.

- Apartado 04 – ARRAYS

En este apartado se han declarado 3 arrays, 2 de Floats y uno de Integer. Seguidamente se han inicializado todos con un valor por defecto de 0.0 en el caso de Float y 0 en el caso de Integer.

Por último se han mostrado por consola todos las posiciones de los arrays para comprobar que, efectivamente, estaban inicializadas a cero.

Juan Pedro Ramos Gómez
Javier Valero Moreno

- Apartado 05 – CADENAS

En este apartado vamos a trabajar con constantes. Definimos la constante ‘cadena’ de tipo String que almacena el valor “TIEMPO REAL”. Seguidamente en el ‘begin’ indicamos que se muestre por consola.

- Apartado 06 – REGISTROS

En este apartado vamos a trabajar con dependencias de ‘types’.

En primer lugar se declaran los ‘types’ Dia y Mes con sus rangos respectivos (1..31 y 1..12 respectivamente).

Despues se crean dos ‘types’ mas:

- Fecha, que contiene un Dia, Mes y un Integer.
- Datos_Personales, que contiene Nombre (String), Apellidos (String) y Fecha_Nacimiento (type Fecha)

Seguidamente se inicializa con una serie de valores concretos, por ejemplo, en este caso los datos del compañero Javier.

Finalmente en el ‘begin’ se imprime por consola los valores asignados accediendo a ellos a través de ‘type’ Datos_Personales.

- Apartado 07 – TIPOS DINAMICOS

En este apartado se da un giro de tuerca al apartado anterior. La principal diferencia es que se ha declarado un ‘type’ Nodo con el fin de crear una lista enlazada de ‘types’ Datos_Personales indicando cual es el valor (Dato) y cual es el siguiente nodo al que acceder (Next_E)

En el ‘begin’ se inicializa el ‘type’ Nodo asignándole los mismo datos que en el ejercicio anterior. Se puede apreciar como debemos dar un paso mas para el acceso a los datos.

Ejemplo: (Antes) Mis_Datos.Nombre -> (Ahora) New_Node.Dato.Nombre

Ejercicio 02.- Estructuras de Control

- Apartado 01

Declaramos un ‘type’ String, Char e Integer.

En el ‘begin’ un for leerá la estructura carácter a carácter y, en función del valor, devolverá una cadena de texto.

Las opciones posibles serían:

- “Opcion 1” si el ‘type’ Char vale ‘A’ o ‘B’
- “Opcion 2” para ‘C’ o ‘D’
- “Opcion 3” para ‘F’

Y “Otra opción” para el resto de casos posibles.

Ejercicio 03.- Ficheros

- Apartado 01

En primer lugar se declaran los ‘types’ custom que van a dar soporte al ejercicio:

- Matriz (array bidimensional)

Ademas tenemos Integers y File_Type.

Abrimos el archivo descargado “input.txt” y dejamos el stream abierto.

Se crea el archivo de salida con nombre “output.txt”.

Un bucle ‘while’ lee el archivo hasta el final realizando la traspuesta de la matriz que contiene ‘input.txt’ y almacenándola finalmente en ‘output.txt’ mediante un bucle for.

Ejercicio 04.- Ocultación de información

- Apartado 01

En este ejercicio hemos usado las funciones de los paquetes “Cola” y “Pila” para realizar operaciones complejas de manejo de estructuras con un nivel de abstracción alto.

Las funciones no están implementadas en el propio paquete, aun asi, eso es transparente al usuario y aumenta la seguridad de nuestro código fuente.

En primer lugar se declara la variable ‘el’ de tipo Integer.

Acto seguido se utiliza la función “Poner()” para añadirla a la estructura de datos y después y con un bucle for, la función “Quitar()” para sustraerlas. Luego se muestra el resultado por consola.

Ejercicio 05.- Sobrecarga de operadores

- Apartado 01

En este ejercicio nos dan un paquete llamado “números_complejos” el cual está incompleto. Lo que se pide en el ejercicio es que se complete le paquete y después se cree un archivo de prueba donde se utilicen las funciones programadas para comprobar el funcionamiento.