****

**Laboratorio 7**

**Registros**

**Objetivos**

* Conocer el funcionamiento de los registros.
* Resolver problemas básicos utilizando registros.

**Prelaboratorio**

Un **registro (RECORD)**, al igual que los arrays, es una estructura de datos estáticas, por lo que el espacio calculado en memoria se calcula en tiempo de compilación. Este tipo de datos estructurados permite superar alguna de las limitaciones observadas en los arrays. Puede definirse como:

Un registro es una estructura de datos que está formada por un conjunto finito de elementos heterogéneos. Cada uno de estos elementos se denomina CAMPO.

Es decir, un registro permite definir un conjunto de elementos que pueden ser del mismo tipo o de tipos diferentes y agruparlos en una única estructura. Para cada uno de estos elementos o campos es preciso especificar un identificador y un tipo.

La sintaxis que permite declarar un registro es:

**TYPE**

**Tipo\_registro = RECORD**

**campo1 : tipo1;**

**campo2 : tipo2;**

**…**

**Campok: tipok;**

**END;**

Varios campos pueden aparecer en la misma línea, separados por una coma, si son del mismo tipo. Al igual que en el caso de los arrays, puede declararse un campo de cualquier tipo definible en Pascal. Es decir, será posible definir registros que contengan registros, y registros que contengan arrays.

La declaración de un registro, al igual que en el caso de otros tipos de datos, puede realizarse en la sección de declaración de tipos o en la sección de declaración de variables de un programa. De manera similar a los otros casos, suele ser conveniente asignarles un identificador de tipo en la sección de tipos, y utilizar ese identificador para declarar variables donde sea necesario.

Para comprender el funcionamiento de un registro, se va a definir incrementalmente una estructura de datos que permita almacenar la siguiente información relativa a una persona:

Fecha\_nacimiento: RECORD

Dia: integer;

Mes: string;

year: integer;

END;

**Tipo\_persona** = RECORD

Nombre, apellidos : STRING[50];

Edad:integer;

Sexo: char; {podría ser un enumerado, un booleano, etc.}

Fecha\_nac: Fecha\_nacimiento;

END;

VAR

Sonia, Antonio, Jorge, Maria, Ismael, Sergio, David: Tipo\_persona;



Como puede comprobarse en la figura anterior, ambas estructuras de datos permiten almacenar la información necesaria. Esta estructura puede complicarse tanto como se desee. Por ejemplo, si se desea añadir un campo nuevo: Lista\_amigos, formado por un array de Max\_amigos de Tipo persona, que permitiría almacenar al conjunto de personas que son amigas o conocidas, de una en particular, la estructura modificada quedaría de la siguiente forma:

Program agenda;

uses crt;

CONST Max\_amigos = 100;

**TYPE**

Tipo\_Fecha = Record

Dia : integer;

Mes: string;

year: integer;

END;

Tipo\_persona = RECORD

Nombre, Apellidos : STRING [50];

Sexo ; char;

Cumple :Tipo\_Fecha;

END;

Agenda\_amigos=RECORD

Nombre, Apellidos : STRING [50];

Lista\_amigos: ARRAY [1..Max\_amigos] OF Tipo\_persona;

END;

VAR

Sonia\_amigos :Agenda\_amigos;

BEGIN

END.

**OPERACIONES SOBRE REGISTROS**

Puede definirse las siguientes operaciones básicas sobre las estructuras de datos de tipo registro:

1. Acceso a los campos de un registro.

2. Asignación entre registros.

3. Paso de registros como parámetros.

4. Lectura/escritura de registros.

1) **Acceso a los campos de un registro**

Cuando se utilizaban arrays, era necesario utilizar un índice para acceder directamente al elemento deseado del array. De la misma forma, puede accederse directamente a cada uno de los campos que forman el registro. Para ello se utilizará una **variable selectora** del campo correspondiente. Esta variable selectora, o simplemente **selector**, está formado por el operado punto (**.**) seguido del identificador del campo al que se desea acceder. De forma genérica, el acceso a un campo particular de un registro puede representarse como:

nombreRegistro.**identificadorCampo ------>** Selector

program ejemplo;

uses crt;

type

datos\_estudiante=RECORD

nombre: string[20];

edad: integer;

sexo: char;

end;

var

registro:datos\_estudiante;

procedure registrar;

begin

writeln('Introduzca su nombre');

readln(registro.nombre);

writeln('Introduzca su edad');

readln(registro.edad);

writeln('Introduzca su genero');

readln(registro.sexo);

end;

procedure mostrar;

begin

writeln('......................................');

writeln('Su nombre es ', registro.nombre);

writeln('Su edad es ', registro.edad);

writeln('Su genero es ', registro.sexo);

writeln('......................................');

end;

begin

registrar;

mostrar;

readkey;

end.

Puesto que un verdadero registro puede contener otras estructuras de datos de forma anidada (como arrays o registros), una vez utilizada el selector para acceder a un campo estructurado, puede utilizarse nuevamente para acceder a niveles más internos de la estructura de datos.

Por ejemplo :



type

**reg\_nombre\_completo = record**

nombre: string[50];

apellido : string[70];

**end;**

**reg\_direccion = record**

manzana,casa: integer;

**end;**

**reg\_direccion\_completa=record**

**direccion: rge\_direccion;**

ciudad,estado: string[30];

codigo\_postal: string[4];

**end;**

**reg\_empleado=record**

nombre\_completo: reg\_nombre\_completo ;

cedula: string[8];

edad: integer;

**direccion: reg\_direccion\_completa ;**

**end;**

**var**

**empleado:reg\_empleado;**

**Otro ejemplo de registro anidado**

program ejemploanidado;

uses crt;

type

fecha=record

Dia,year:integer;

Mes: string;

end;

datos\_estudiante=record

nombre: string[20];

edad: integer;

sexo:char;

fechas:fecha;

end;

var

registro:datos\_estudiante;

procedure registrar;

begin

with registro do

begin

writeln('Introduzca su nombre');

readln(nombre);

writeln('Introduzca su edad');

readln(edad);

writeln('Introduzca su genero');

readln(sexo);

writeln('Introduzca su fecha de nacimiento');

readln(fechas.dia);

readln(fechas.mes);

readln(fechas.year);

end;

end;

procedure mostrar;

begin

with registro do

begin

writeln('......................................');

writeln('Su nombre es ', nombre);

writeln('Su edad es ', edad);

writeln('Su genero es ', sexo);

writeln('Su fecha de nacimiento es ', fechas.dia, '- ', fechas.mes, '- ',fechas.year);

writeln('......................................');

end;

end;

begin

registrar;

mostrar;

readkey;

end.

2) **Asignación entre registros**

De forma similar a como sucede con los arrays, existen algunas operaciones que pueden ser realizadas entre dos registros. La copia de información entre dos registros es la operación más básica. Cuando se desea copiar los datos almacenados de un registro a otro, existen dos posibilidades. La primera consiste en copiar, campo a campo, cada uno de los datos del registro origen en el registro destino. La segunda, sólo aplicable si ambos registros **son del mismo tipo** o de tipos compatibles, consiste en asignar directamente el valor del registro origen sobre el registro destino.

3) **Paso de registros como parámetros.**

Dado que una función no puede devolver un tipo estructurado no podrá utilizarse para devolver un registro. Un registro puede pasarse como parámetros a una función o procedimiento siempre y cuando los parámetros formales definidos en la cabecera de estos subprogramas tengan el mismo tipo que los correspondientes parámetros actuales. Por ejemplo, las siguientes declaraciones son válidas en Pascal.

VAR

Sonia, Antonio, Maria, David, :Tipo\_personas;

PROCEDURE uno (a, b :Tipo\_persona; VAR c : real);

….

PROCEDURE dos (a :Tipo\_persona; VAR b,c : Tipo\_persona) ;

…..

BEGIN

Uno (Sonia, Antonio, 3.45) ;

Dos (Maria, Sonia, David );

END

4) **Lectura/escritura de registros**

No es posible leer ni escribir directamente todos los campos de un registro. Para leer o escribir los campos de un registro, debe accederse de forma individual a cada uno de ellos; para acceder a los diferentes campos del registro se utilizaran sus correspondientes selectores.

**LABORATORIO**

1. Defina un registro que permita guardar la siguiente información : Cédula, Nombre, Apellido, FechaNacimiento (DDMMAAAA), Sexo, Semestre que cursa, Mensaje, Materia más difícil.

Considere los tipos de datos mas conveniente para cada uno de los campos del registro.

Seguidamente haga un programa en Pascal que permita al usuario llenar la información correspondiente en un registro y posterior a ello visualizarla en pantalla.

2. Elabore un programa en Pascal para registrar diez (10) artículos de una tienda los cuales deben tener la siguiente información: código, nombre, familia, precio y cantidad. Donde la familia es un código numérico entre 1 y 4 que agrupa por tipo de material ( 1 : papelería, 2 : computación, 3 : MAE materiales eléctricos, 4: otros). Luego de registrar la información, en un arreglo, listar por familia cada uno de los productos registrados.

3. Una compañía disquera organiza una encuesta para determinar el éxito de sus canciones. La población encuestada es dividida en cuatro categorías de acuerdo al sexo y a la edad (Por ejemplo: en personas de 20 años o menos y mayores de 20 años). A cada persona se le piden sus cinco canciones favoritas, las canciones son identificadas por números del 1 al N. Los resultados de la encuesta son tomados de la entrada estándar. Cada elemento leído representa un encuestado y contiene su cédula, nombre, apellido, sexo, edad y sus cinco canciones favoritas en orden de preferencia. Se desea que Ud. diseñe una estructura de datos a través de la sentencia RECORD en el lenguaje de programación Pascal para almacenar dicha información.

**POST LABORATORIO**

**Solo debe resolver la cantidad de ejercicios cuyo valor sumen 20 puntos.**

NOTA IMPORTANTE : los datos deben leerse en estructuras de registro y de requerir almacenarlos utilizar arreglos..

1. Una compañía dedicada al transporte de pasajeros por vía aérea, desea contratar sus servicios con el fin de automatizar el proceso de reservación de sus vuelos. Para ello Ud. deberá crear una estructura de datos en el lenguaje Pascal que permita almacenar dicha información, asumiendo libremente y dentro de la lógica común los campos que debe contener la estructura de registro y escribir el registro por pantalla. ( 5 ptos)
2. En una ciudad se conocen los siguientes datos de los ciudadanos: Cédula de Identidad, Nombres, Apellidos, Edad, Sexo, Estado Civil y Nacionalidad. Para los hombres, además se conoce si tiene pies planos o no y si prestó o está prestando actualmente el Servicio Militar. . Se desea que Ud. implemente una estructura de datos en el lenguaje Pascal para almacenar esta información. Además se pide un programa que permita las siguientes operaciones:
3. Listar la cédula, nombres y apellidos de los ciudadanos que puede votar en las próximas elecciones. ( 5 ptos)
4. Listar la cédula, nombres y apellidos, de aquellos ciudadanos que pueden prestar servicio militar. **( 5 ptos)**
5. Diseñe una estructura de datos y un programa en el lenguaje de programación Pascal, para almacenar y manejar la información concerniente a los resultados de los exámenes recuperativos. El programa debe:
6. Tomar como entrada los Cédula de Identidad, Nombre y Apellido de los alumnos, la carrera y la información de los objetivos evaluados , y para cada uno de los objetivos indicar si fue Aprobado o No. **(5 ptos)**
7. El Banco LATINO, desea automatizar todas sus operaciones bancarias. Por tal razón desea contratar sus servicios, con la finalidad que Ud. diseñe una Estructura de Datos en el lenguaje Pascal que permita manejar de una manera fácil y eficiente todas sus operaciones, considerando los siguientes requerimientos:

* El banco maneja básicamente tres (3) tipos de cuentas cuentas de ahorro, corrientes y de activos líquidos; además por cada una de ellas deberá llevarse la tasa de rendimiento, en %, correspondiente
* Para cada uno de los clientes se deberá conocer la siguiente información: Cedula de Identidad , Número de la Cuenta, Tipo de Cuenta, Nombre, Dirección, Teléfono y Saldo en la Cuenta.
* Un cliente podrá realizar transacciones sobre una cuenta, de las cuales debe registrarse : Nro. de la Transacción, Fecha de la misma, Número de Cuenta Involucrada, Monto en Bolívares y el Número de la caja donde fue realizada. Elaborar un programa que permita registrar la data anteriormente descrita y sobre ella generar un los siguientes reportes o consultas :

1. Lista de todos los clientes con Cuenta Corriente, ordenados alfabéticamente en orden creciente.
2. Lista de todas las transacciones que se hicieron sobre una cuenta de ahorros dada en un período de fecha también dado. **(10 ptos**)

**Instrucciones para el envío de la tarea :**

Crear una carpeta comprimida que incluya:

* Un archivo .doc con los algoritmos de los problemas propuesto en el Postlaboratorio y con las capturas de pantalla de las corridas de los programas respectivamente.
* Los archivos .pas de los problemas resuelto (un archivo .pas por cada programa)

Además debe enviar la HOJA DE CONTROL DE PRACTICAS con la autoevaluación de la práctica de la semana.

Adjuntar archivos a la tarea correspondiente en Módulo 7