TALLER DE PROGRAMACIÓN

Clase 0
Programación Pascal

Temas de la clase

1. Introducción a la Programación Pascal utilizando **Geany** (se encuentra disponible en la medioteca de **Ideas**).
2. Ejercitación con operaciones de Vector y Lista
3. Resolución de un problema

Introducción a la programación Pascal

1. Pasos para la escritura de un programa
   a. Escritura
   b. Compilación
   c. Ejecución
2. Operación Random

Actividades en Máquina

ACTIVIDAD 1

a) Copie el siguiente programa.

```
program NumAleatorio;
var ale: integer;
begin
   randomize;
   ale := random (100);
   writeln ('El número aleatorio generado es: ', ale);
   readln;
end.
```

b) Compile y ejecute el programa descargado

c) Responda ¿Qué hace el programa?

Actividades en Máquina

ACTIVIDAD 1 (continuación)

Modifique el programa incorporando:

d) Un módulo que imprima 20 números aleatorios.

e) Un módulo que imprima N números aleatorios en el rango (A,B), donde N, A y B son números enteros que se leen por teclado y se reciben como parámetros.

e) Un módulo que imprima números aleatorios en el rango (A,B) hasta que se genere un valor igual a F, el cual no debe imprimirse. F, A y B son números enteros que se leen por teclado y se reciben como parámetros.

Actividades en Máquina

ACTIVIDAD 2: Crear un nuevo archivo `ProgramaVectores.pas`

a) Implemente un módulo **CargarVector** que cree un vector de enteros con a lo sumo 50 valores aleatorios. Los valores, generados aleatoriamente (entre un mínimo y máximo recibidos por parámetro), deben ser almacenados en el vector en el mismo orden que se generaron, hasta que se genere el cero.

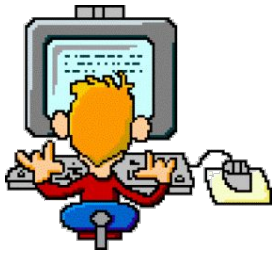
b) Implemente un módulo **ImprimirVector** que reciba el vector generado en a) e imprima todos los valores del vector en el mismo orden que están almacenados. Qué cambiaría para imprimir en orden inverso?

c) Escriba el cuerpo principal que invoque a los módulos ya implementados.

Actividades en Máquina

ACTIVIDAD 3: Crear un archivo `ProgramaListas.pas`

a) Implemente un módulo **CargarLista** que cree una lista de enteros y le agregue valores aleatorios entre el 100 y 150, hasta que se genere el 120.
b) Implemente un módulo **ImprimirLista** que reciba una lista generada en a) e imprima todos los valores de la lista en el mismo orden que están almacenados.
c) Implemente un módulo **BuscarElemento** que reciba la lista generada en a) y un valor entero y retorne true si el valor se encuentra en la lista y false en caso contrario.
d) Invocar desde el programa principal a los módulos implementados para crear una lista, mostrar todos sus elementos y determinar si un valor leído por teclado se encuentra o no en la lista.



Actividades en Máquina

ACTIVIDAD 4: Crear un archivo `ProgramaListasOrdenadas.pas`

a) Implemente un módulo **CargarListaOrdenada** que cree una lista de enteros y le agregue valores aleatorios entre el 100 y 150, hasta que se genere el 120. Los valores dentro de la lista deben quedar ordenados de menor a mayor.

b) Reutilice el módulo **ImprimirLista** que reciba una lista generada en a) e imprima todos los valores de la lista en el mismo orden que están almacenados.

c) Implemente un módulo **BuscarElementoOrdenado** que reciba la lista generada en a) y un valor entero y retorne true si el valor se encuentra en la lista y false en caso contrario.

d) Invocar desde el programa principal a los módulos implementados para crear una lista ordenada, mostrar todos sus elementos y determinar si un valor leído por teclado se encuentra o no en la lista.

Actividades en Máquina

ACTIVIDAD 5: Crear un archivo `gimnasio.pas`

Un gimnasio necesita procesar las asistencias de sus clientes. Cada asistencia tiene día, mes, año, número de cliente (entre 1 y 500) y la actividad realizada (valor entre 1 y 5).

a) Implemente un módulo que retorne una lista de asistencias de clientes un gimnasio. Las asistencias dentro de la lista deben quedar ordenadas de menor a mayor por número de cliente. Generar aleatoriamente los valores hasta generar un valor cero para el número de cliente.

b) Implemente un módulo que reciba la lista generada en a) e imprima todos los valores de la lista en el mismo orden que están almacenados.

c) Implemente un módulo que reciba la lista generada en a) y un número de cliente y retorne la cantidad de asistencias del cliente recibido. Mostrar el resultado desde el programa principal.

d) Implemente un módulo que reciba la lista generada en a) y retorne la actividad con mayor cantidad de asistencias. Mostrar el resultado desde el programa principal