UNER Facultad de Ciencias de la Administración

Algoritmos y Programación

Subprogramas

Prof. Miguel A. Fernández

1

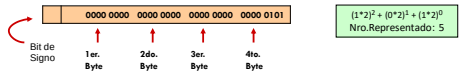
Abstracción

- Mecanismo fundamental del razonamiento humano para manejar la complejidad de las cosas.
- Consiste en prestar atención solamente a los aspectos importantes de un tema dejando de lado los detalles que pueden considerarse secundarios.
- **Se pueden Clasificar:**
  - Aplicada a las operaciones necesarias para resolver un problema, en cuyo caso se denomina **abstracción funcional o abstracción de procesos**
  - Aplicada a la representación de los datos necesarios para resolver un problema, en cuyo caso se denomina **abstracción de datos**

Universidad Nacional de Entre Ríos

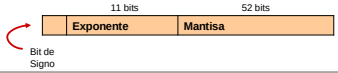
Abstracción

Representación de una variable entera (de 32 bits):



Representación de una variable real (punto flotante) de 32 bits:

Representación de una variable real de 64 bits:



Universidad Nacional de Entre Ríos

Abstracción de Procesos

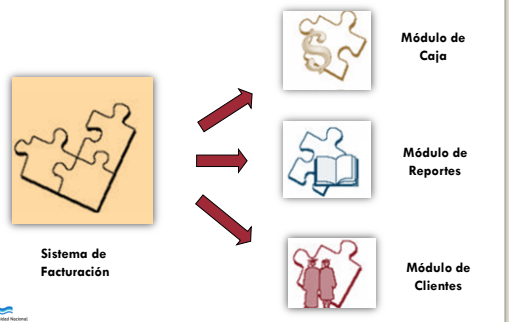
Modularización

Veamos algunas maneras de definir este término:

1. Significa dividir un problema en partes funcionalmente independientes que encapsulen operaciones y datos.
2. Significa descomponer un programa en un número pequeño de abstracciones coherentes que pertenecen al dominio del problema y ocultan la complejidad interna.
3. Es dividir un programa en módulos o partes que pueden ser compilados separadamente o nó, pero existiendo conexiones o acoplamiento entre los módulos.

Universidad Nacional de Entre Ríos

Modularización



Universidad Nacional de Entre Ríos

Modularización

VENTAJAS:

1. Mayor Productividad:

Al dividir un sistema de software en módulos funcionalmente independientes, un equipo de desarrollo puede trabajar simultáneamente en varios módulos Incrementando así la productividad (es decir reduciendo el tiempo de desarrollo global del sistema).

Fuente: http://www.monografias.com/trabajos106/modularizacion/modularizacion2.shtml

Universidad Nacional de Entre Ríos

Modularización

Sistema de Turnos Médicos

- Gestión de Pacientes
  - Alta de Pacientes
  - Baja de Pacientes
  - Consulta/ Modifica Datos Pacientes
- Gestión de Turnos
  - Reserva de Turnos
  - Anulación de Turnos
  - Consulta de Turnos

También un sistema puede visualizarse como un esquema donde cada módulo es ejecutado por un módulo "master", según los requerimientos de proceso.

Modularización

VENTAJAS:

2. Reusabilidad:

Es un objetivo fundamental de la Ingeniería de software. Significa poder contar con la posibilidad de utilizar en forma repetida un producto de software ya desarrollado.

- Gestión de Clientes
  - Alta de Clientes
  - Baja de Clientes
  - Consulta/ Modifica Datos Clientes

➡ **Puede utilizarse para cualquier otro Sistema.**

Modularización

VENTAJAS:

3. Facilidad de Mantenimiento:

La división del problema en módulos permite aislar más fácilmente los posibles errores que se puedan producir. Esto implica menor tiempo de corrección.

¡ No puedo dar de Baja un cliente !

Sólo Afecta: ➡

- Gestión de Clientes
  - Alta de Clientes
  - Baja de Clientes
  - Consulta/ Modifica Datos Clientes

Modularización

VENTAJAS:

4. Facilidad de Crecimiento del Sistema:

A menudo es necesario actualizar los sistemas, incorporando nuevas funcionalidades que surgen de nuevos requerimientos.

Un desarrollo modular permite disminuir los riesgos y costos que surgen del hecho de incorporar nuevas prestaciones a un Sistema en funcionamiento.

Modularización

Sistema de Turnos Médicos

- Gestión de Pacientes
  - Alta de Pacientes
  - Baja de Pacientes
  - Consulta/ Modifica Datos Pacientes
- Gestión de Turnos
  - Reserva de Turnos
  - Anulación de Turnos
  - Consulta de Turnos
- Gestión de Cobranzas
  - Generar Facturas
  - Anular Facturas
  - Impresión Reportes

Modularización

VENTAJAS - Resumen:

1. Mayor Productividad
2. Reusabilidad
3. Facilidad de Mantenimiento
4. Facilidad de Crecimiento del Sistema
5. Mayor Legibilidad

Programación Modular

Forma de programación que consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más legible y manejable.

Es una evolución de la programación estructurada para solucionar problemas de programación más grandes y complejos.

PROGRAMACIÓN MODULAR

Así, un problema complejo debe ser dividido en varios subproblemas más simples, y estos a su vez en otros Subproblemas más simples, hasta obtener subproblemas lo suficientemente simples como para poder ser resueltos fácilmente con a lgún lenguaje de programación.

Ésta técnica se llama refinamiento sucesivo, divide y vencerás ó análisis descendente (Top-Down).

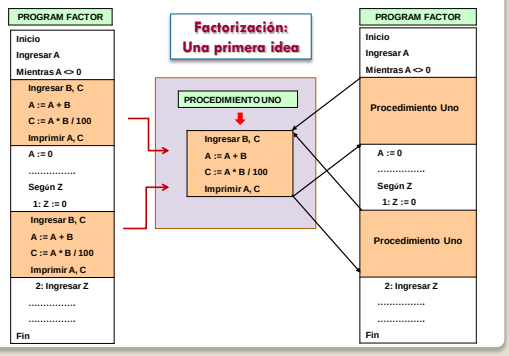
Modularización

La modularización se puede realizar:

a) Dividiendo un programa en partes mas pequeñas (módulos, rutinas, funciones, **procedimientos**) que forman parte del mismo programa.

b) Dividiendo un problema en partes separadas (subprogramas) que se compilan por separados conformando cada uno de ellos una unidad.

Modularización

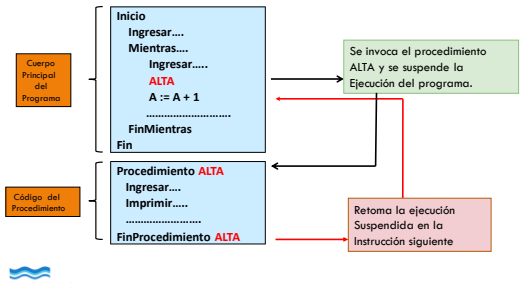


Modularización

Características generales:

- El procedimiento es invocado en forma explícita.
- Cada procedimiento tiene un solo punto de entrada.
- La ejecución de la unidad que llama a un procedimiento es suspendida durante la ejecución del procedimiento.
- El control siempre retorna a la instrucción siguiente a la invocación cuando el procedimiento finaliza.

Subprogramas



Subprogramas

- Son construcciones de bloques (partes) de programas y constituyen uno de los conceptos mas importantes en el diseño de los lenguajes de programación.
  - Centrarse en el "que" para posteriormente ver el "como".

Categorías:

Procedimientos.

Conjunto de sentencias que se computan y son convocadas por un simple llamado o sentencia "call"

Funciones.

Pero este tema lo veremos mas adelante

Subprogramas

EJEMPLO:

Se requiere procesar los datos de las inasistencias del personal de una empresa durante un semestre a fin de proceder a la liquidación de una bonificación. La empresa tiene 80 empleados, cuyos números de legajos están codificados numéricamente en forma correlativa a partir del número 1.

Se requiere codificar un programa que contemple:

a) Ingresar los nombres de los 80 empleados

b) Procesar las asistencias en que incurrieron los empleados, para lo cual se ingresa por teclado:
- Número de legajo
- Mes en que se verificó la/s inasistencia/s
- Cantidad de inasistencia del mes

c) Imprimir un listado que contenga

Nombre Empleado – Cantidad de Inasistencias en el semestre

Universidad Nacional de Entre Ríos

Subprogramas

¿ Cómo lo resolvíamos ?

```
Program Busco
Type
   PERSO = Record
         NOM : String;
         INAS :  Integer;
   end;
   Empleados = array [1..80] of PERSO;
Var
    EMP : Empleados;
    i, NL,CAIN  : Integer;
    ESTA : Boolean;
```

```
Inicio
  Para i = 1, 80, 1
     EMP(i).INAS := 0
  Finpara
  Mostrar "Ingreso Nombres Empleados"
  Ingresar NL
  Mientras NL  <> 0
     Ingresar EMP(NL).NOM
     Ingresar NL
Finmientras
Mostrar "Proceso de Inasistencias"
Ingresar NL
Mientras NL  <> 0
   Ingresar CAIN
   EMP(NL).INAS := EMP(NL).INAS + CAIN
   Ingresar NL
Finmientras
Mostrar "Impresión de Listado"
Para i = 1, 80, 1
   Imprimir EMP(i).NOM, EMP(i).INAS
FinPara
Fin
```

Universidad Nacional de Entre Ríos

Subprogramas

Módulo Principal

```
Inicio
Para i = 1, 80, 1
          EMP(i).INAS = 0
Finpara
Mostrar "Seleccione alguna Opción"
          " 1 = Ingresar Nombres "
          " 2 = Procesar Inasistencias"
          " 3 = Imprimir listado"
          " 0 = Fin de Programa "
Ing. OPCION
Mientras OPCION <>  0
   Según OPCION
          1 : NOMBRE
          2 : INASISTENCIAS
          3 : LISTADO
          Otro Valor:
            Mostrar: "Código Incorrecto"
   Finsegún
   Ingresar OPCION
Finmientras
Fin
```

```
Procedimiento NOMBRE
Mostrar "Ingrese Nombres Empleados,  FIN=0 "
Ingresar NL
Mientras NL  <>  0
     Ingresar EMP(NL).NOM
     Ingresar NL
Finmientras
FinProcedimiento NOMBRE
```

```
Procedimiento INASISTENCIAS
Mostrar "Proceso de Inasistencias, FIN=0"
Ingresar NL
Mientras NL  <>  0
     Ingresar CAIN
     EMP(NL).INAS := EMP(NL).INAS + CAIN
     Ingresar NL
Finmientras
Fin-Procedimiento INASISTENCIAS
```

```
Procedimiento LISTADO
Mostrar "Impresión de Listado"
Para i = 1, 80, 1
        Imprimir EMP(i).NOM,  EMP(i).INAS
FinPara
Fin-Procedimiento LISTADO
```

Subprogramas

Módulo Principal

```
Inicio
Para i = 1, 80, 1
          EMP(i).INAS = 0
Finpara
MENU
Mientras OPCION <>  0
   Según OPCION
          1 : NOMBRE
          2 : INASISTENCIAS
          3 : LISTADO
          Otro Valor:
            Mostrar: "Código Incorrecto"
   Finsegún
   MENU
Finmientras
Fin
```

```
Procedimiento MENU
Mostrar "Seleccione alguna Opción"
          " 1 = Ingresar Nombres "
          " 2 = Procesar Inasistencias"
          " 3 = Imprimir listado"
          " 0 = Fin de Programa "
Ingresar OPCION
FinProcedimiento MENU
```

```
Procedimiento NOMBRE
Mostrar "Ingrese Nombres Empleados,  FIN=0"
Ingresar NL
Mientras NL  <>  0
     Ingresar EMP(NL).NOM
     Ingresar NL
Finmientras
FinProcedimiento NOMBRE
```

```
Procedimiento INASISTENCIAS
Mostrar "Proceso de Inasistencias, FIN=0"
Ingresar NL
Mientras NL  <>  0
     Ingresar CAIN
     EMP(NL).INAS := EMP(NL).INAS + CAIN
     Ingresar NL
Finmientras
Fin-Procedimiento INASISTENCIAS
```

```
Procedimiento LISTADO
Mostrar "Impresión de Listado"
Para i = 1, 80, 1
        Imprimir EMP(i).NOM,  EMP(i).INAS
FinPara
Fin-Procedimiento LISTADO
```

Subprogramas

Definición de Objetos de Datos:

¡ Es posible especificar declaraciones en un procedimiento !

Ejemplo:

```
Program GRANDE
 Var
     N1, N2 : Integer;
 Begin
     Calculo
 Fin
```

Variables Globales
Visibles en todo el programa y todos los procedimientos.

```
Procedure Calculo
Var
     Res : Integer;
Begin
  Ingresar  N1, N2
  RES := N1  + N2
  Imprimir RES
Fin-Procedure Calculo
```

Variables Locales
Visibles solamente en el procedimiento donde está definida.

Universidad Nacional de Entre Ríos

Subprogramas

Luego:

```
Program GRANDE
 Var
      N1, N2 : Integer;
 Begin
     Calculo
     RES := RES + 1
 Fin
```

¡ ERROR !
Variable no reconocida

```
Procedure Calculo
Var
      Res : Integer;
 Begin
     Ingresar  N1, N2
     RES := N1 + N2
     Imprimir RES
Fin-Procedure Calculo
```

Universidad Nacional de Entre Ríos

Subprogramas

En Resumen:

Las variables pueden ser:

- **Variables locales.**
  - Se declaran dentro del módulo (procedimiento o función).
  - Se utilizan solamente dentro del módulo.
  - Están vivas solamente durante la ejecución del módulo.
  - Se crean en el momento en que el módulo es invocado.
- **Variables globales:**
  - Se declaran fuera de los módulos/procedimientos.
  - Pueden ser usadas por el cuerpo principal y por cualquier módulo.

Subprogramas

Definición en Pascal

Se codifica primero los procedimientos antes que el cuerpo principal

Ejemplo:

```
Program GRANDE
Var
    N1, N2 : Integer;
Procedure Calculo
Var
    Res : Integer;
Begin
  Ingresar  N1, N2
  RES := N1 + N2
  Imprimir RES
Fin-Procedure Calculo
Begin
  Imprimir "Es un ejemplo"
  Calculo
End.
```

Definición de variables globales y tipos de usuarios si los hubiere

Procedimiento

Cuerpo Principal

Subprogramas

¿ Qué valor Imprime ?

```
Program Grande;
Var
   Todo : Integer;
Procedure Uno;
  Begin
    Todo := 27;
    write Todo;
    Todo := 50;
  end;
Begin
  Todo := 0;
  write Todo;
  Uno;
  write Todo;
End.
```

? ? ?

```
Program Grande;
Var
   Todo : Integer;
Procedure Uno;
Var
   Todo : Integer;
  Begin
    Todo := 27;
    write Todo;
    Todo := 50;
  end;
Begin
  Todo := 0;
  write Todo;
  Uno;
  write Todo;
End.
```

? ? ?

Arreglos Compuestos - Registros

¡ TIEMPO DE TRABAJO !

Subprogramas - Ejercicio

Una mutual de empleados debe procesar los datos de sus asociados para lo cual, el analista ha definido la codificación de los siguientes procesos:

a) Se deben ingresar los siguientes datos de todos los asociados:
- Número de Asociado (ordenados en forma correlativa a partir del número 1 al 255)
- Nombre del Titular
- Número de Documento
- Saldo que debe por compras realizadas en distintos comercios y préstamos obtenidos.

b) Ingresar las operaciones que se realizaron con fecha posterior al saldo registrado en el punto anterior y actualizar el saldo que debe.
- Número de Asociado
- Fecha de la Operación
- Código de operación [1=Pago, 2=Compra 3=Cuota por préstamo que otorga la mutual]
- Importe

c) Imprimir un listado de todos los clientes ordenado por Número de Asociado que contenga:
Número de Asociado - Nombre del Asociado - Saldo Actual

d) Mostrar por pantalla un cuadro estadístico con los siguientes datos:
- Importe total adeudado por los Asociados
- Nombre y número de documento del Asociado que mas importe adeuda
- Nombre del Asociado que paga la mayor cuota por importe prestado por la mutual.

Se debe codificar un módulo por cada procesos, los que deben ser invocados por un menú de opciones.

Subprogramas

FIN

MUCHAS GRACIAS