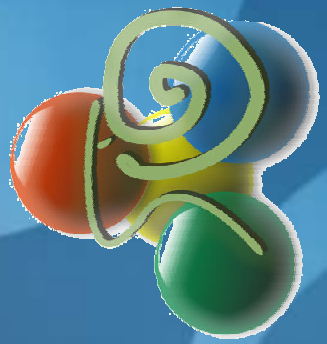
Célula Académica
UABC-Live .net



Universidad Autónoma de Baja California
Facultad de Ciencias Químicas e Ingeniería

http://uabc-live-net.spaces.live.com/

Sesión No. 2

Fundamentos de la Programación

Expositores: Mario Ceceña Acosta (lemariee@hotmail.com)

Juan de Dios Pérez Camacho (juanddpc@hotmail.com)

Fecha: 28 de septiembre de 2006

Programa Microsoft Desarrollador Cinco Estrellas

Estrella 0

Objetivo

Mostrar los fundamentos de la programación a través de ejemplos y prácticas utilizadas cotidianamente en el desarrollo de aplicaciones

Prerrequisitos

- El presente curso asumirá conocimientos básicos de
  - Computadora
  - Dispositivos de Entrada/Salida
  - Organización Física de una computadora (CPU, Memoria)
  - Sistemas Operativos

Temas a Tratar (1/2)

- El Software
- Lenguajes de programación
- Resolución de problemas con computadora
- Entorno de programación
- Tipos de Datos
- Variables y Constantes
- Sentencias
- Operadores y Expresiones

Temas a Tratar (2/2)

- Estructuras Básicas de Control
- Procedimientos y Funciones
- Visibilidad de variables
- Bibliotecas
- Arrays
- El Estilo de Programación

El Software

- Las operaciones que debe realizar el hardware son especificadas con una lista de instrucciones, llamadas programas o software.
- Dos grandes grupos de software
  - Software del Sistema
    - Indispensable para que la máquina funcione y poder escribir programas de aplicación
  - Software de Aplicación
    - Realizan tareas concretas que tienen utilidad para ciertos usuarios

Lenguajes de Programación (1/2)

- Lenguajes utilizados para escribir programas de computadoras que puedan ser entendidos por ellas
- Se clasifican en tres grandes categorías
  - lenguajes de máquina
    - instrucciones directamente entendibles por la computadora (lenguaje binario)
  - lenguajes de bajo nivel
    - Proveen un juego de instrucciones más comprensibles por los humanos
  - lenguajes de alto nivel

Lenguajes de Programación (2/2)

- Lenguajes de alto nivel
  - Utilizan instrucciones escritas con palabras similares a los lenguajes humanos
  - Son independientes de la máquina en la que se ejecutan
  - Necesitan ser traducidos a instrucciones en lenguaje máquina (Compilación)
- Existen diversos tipos
  - Estructurados
  - Orientados a Objetos
  - Declarativos
  - Funcionales

Resolución de problemas con computadora

- El proceso de diseñar un programa es, esencialmente, un proceso creativo.
- Sin embargo, hay una serie de pasos comunes a seguir:
  - Análisis del problema
  - Diseño del algoritmo solución
  - Codificación
  - Compilación y Ejecución
  - Verificación
  - Depuración
  - Documentación

Entorno de Programación

- También conocidos como IDEs
- Herramienta esencial a la hora de desarrollar software
- Incluye
  - Editor
  - Intérprete o Compilador
  - Depurador
  - Ayuda en línea

Tipos de Datos

- Datos: piezas de información con las que un programa trabaja
- Cada dato tiene asociado un único *Tipo*
- El Tipo de Dato determina la naturaleza del conjunto de valores que un dato puede tomar
- Ejemplos:
  - Número Entero
  - Número Real
  - Cadena de Caracteres
  - Valor Lógico (Verdadero o Falso)

Variables y Constantes

- Existen dos grupos principales de datos
  - Constantes: su valor no puede cambiar durante la ejecución de un programa
  - Variables: su valor puede cambiar durante la ejecución de un programa
- Ambas tienen un nombre y un valor
- Ambas permiten representar mediante un nombre a una posición de memoria que contiene el valor

Sentencias

- Describen acciones algorítmicas que pueden ser ejecutadas
- Se clasifican en
  - Ejecutables / No ejecutables
  - Simples / Estructuradas

Operadores y Expresiones (1/2)

- Sirven para procesar variables y constantes
- Una expresión es un conjunto de datos unidos por operadores que tiene un único resultado
  - Expresiones aritméticas
    - El resultado es un número
    - *a = ((2+6) / 8) * 3*
  - Expresiones lógicas
    - El resultado es un valor verdadero o falso
    - *(a < 10) y (b > 50)*

Operadores y Expresiones (2/2)

- Existen diversos tipos
  - Aritméticos: suma, resta, multiplicación, etc.
  - De relación: igual, mayor, menor, distinto, etc.
  - Lógicos: and, or, not, etc.

Estructuras de Control

- El orden de ejecución de las sentencias de un programa determina su flujo de control
- Las estructuras de control permiten alterar el orden del flujo de control
- Existen dos tipos básicos
  - De Selección
  - De Repetición o Iteración

Estructuras de Control Selectivas (1/2)

- Dirigen el flujo de ejecución según el resultado de evaluación de expresiones
- IF
  - *si* expresion_logica

    entonces hacer acción A

    sino hacer acción B

    fin_si

Estructuras de Control Selectivas (2/2)

- CASE
  - *según_sea* selector *hacer*

    C11,C12,....: sentencia 1

    C21,C22,....: sentencia 2

    .....

    [*sino* sentencia x]

    fin_según

Estructuras de Control Repetitivas (1/3)

- Permiten ejecutar un conjunto de sentencias repetidamente una cierta cantidad de veces o hasta que se cumpla una determinada condición
- El conjunto de sentencias se denomina bucle
- Cada repetición del cuerpo del bucle se denomina iteración

Estructuras de Control Repetitivas (2/3)

- WHILE
  - *mientras* condición *hacer*

    sentencia/s

    .....

    fin_mientras

Estructuras de Control Repetitivas (3/3)

- FOR
  - *desde* variable ← valor_inicial *hasta* valor_final *hacer*

    sentencia/s

    .....

    fin_desde

Procedimientos y Funciones (1/4)

- Descomposición en subprogramas: estrategia para resolver problemas complejos
- Los subprogramas se implementan a través de procedimientos y funciones
  - Compuestos por un grupo de sentencias
  - Se les asigna un nombre
  - Pueden invocarse entre sí utilizando ese nombre
  - Constituyen una unidad de programa

Procedimientos y Funciones (2/4)

- Los procedimientos y funciones se comunican con su invocador a través de parámetros.
- Los parámetros son un medio para pasar información, implementados a través de variables con valor.
- Tipos de parámetro
  - De Entrada: su valor es proporcionado por el invocador antes de llamar al subprograma
  - De Salida: su valor es calculado dentro de un subprograma y devuelto a su invocador

Procedimientos y Funciones (3/4)

- **Ejemplo:**
  - **Definición**

    procedimiento CalcularSuma(parámetro1 entero, parámetro2 entero) devuelve entero

    devolver parámetro1 + parámetro2

    fin_procedimiento
  - **Invocación desde el programa principal u otro subprograma**

    número entero a = 2

    número entero b = 3

    número entero c = CalcularSuma(a,b)

    carácter d = CalcularSuma(a,b) → ERROR

Procedimientos y Funciones (4/4)

- Ventajas de utilizar procedimientos
  - Facilita el diseño descendiente y modular
  - Promueven la reutilización de código
  - Facilita la división de tareas
  - Pueden comprobarse individualmente
  - Pueden encapsularse en bibliotecas independientes

Visibilidad de Variables

- Variable Local:
  - Declarada en un subprograma
  - Sólo está disponible durante el funcionamiento del subprograma
  - Su valor se pierde una vez que el subprograma termina
- Variable Global:
  - Declarada en el programa principal
  - Está disponible en el programa principal y en todos los subprogramas
  - Su valor se pierde una vez que el programa principal termina

Bibliotecas

- Archivo independiente que contiene un conjunto de subprogramas
- Pueden ser incluidas y referenciadas en el desarrollo de múltiples programas
- Facilitan la modularización de un programa
- Desarrollo → Programa Fuente
- Compilación → Programa Objeto
- Link-Edición → Programa Ejecutable

Arrays (Arreglos) (1/3)

- Son estructuras de datos en las que se almacenan un conjunto de datos finitos del mismo tipo
  - Almacenan sus elementos en posiciones de memoria contiguas
  - Tienen un único nombre de variable que representa a todos los elementos
  - Permiten acceso directo o aleatorio a sus elementos individuales
- Los arrays se clasifican en unidimensionales y multidimensionales.

Arrays (Arreglos) (2/3)

- Arrays unidimensionales (Vectores)
  - Número finito de elementos
  - Tamaño Fijo
  - Elementos Homogéneos
  - Se accede a los elementos utilizando el nombre del array y el subíndice específico
- Ejemplo:
  - *salarios(3) Reales* → Nombre del array, de 3 posiciones que contendrán número reales
  - *salarios[1] = 23,4* → Asignación de un valor al primer elemento del array

Arrays (Arreglos) (3/3)

- Arrays multidimensionales
  - Arrays bidimensionales (Matrices o Tablas)
    - Tienen dos índices, uno para filas y otro para columnas
    - Ejemplo:

tabla(3,3) enteros → Declaración de una matriz de 3 por 3

tabla [1][1] = 2 → Elemento de la primer fila y primer columna

tabla [2][3] = 5 → Elemento de la segunda fila y la tercer columna

El estilo de Programación

- Una de las características más importantes de un buen programador
- Un buen estilo facilita la comprensión, corrección y mantenimiento de un programa
- Algunos puntos a tener en cuenta
  - Comentarios
  - Elección de nombres significativos
  - Identación
  - Espacios y Líneas en Blanco
  - Validación usando datos de prueba

Microsoft®