

LABORATORIO DE PROGRAMACIÓN
ORIENTADA A OBJETOS

DRA. MARICELA BRAVO

CBI DIVISION DE
CIENCIAS BASICAS
E INGENIERIA
UAM - Azcapotzalco

UNIVERSIDAD
AUTONOMA
METROPOLITANA
Casa abierta al tiempo **Azcapotzalco**

INTRODUCCIÓN

- **Paradigma de programación**
 - Los paradigmas indican las diversas formas que han sido aceptadas como estilos para programar y para resolver los problemas por medio de una computadora.
- **Paradigmas**
 - Programación Procedimental
 - Programación Funcional
 - Programación Lógica
 - Programación Orientada a Objetos

2

PROGRAMACIÓN PROCEDIMENTAL

- Es considerado el más común
- El programador se concentra en el procesamiento, en el algoritmo requerido para llevar a cabo el cómputo deseado.
- Fortran es el lenguaje de procedimientos original; los lenguajes como C, Basic o Pascal surgieron posteriormente con el mismo paradigma.
- La programación estructura se considera como el componente principal de la programación procedimental.



3

PROGRAMACIÓN PROCEDIMENTAL

```

1 #Module Module1
2
3 Sub Main()
4
5 'Bloque de código donde declaro las variables
6 Dim a As Integer, b As Integer
7 Dim suma, resta, producto, div As Double
8 Dim modulo As Double
9
10 'Bloque de código para realizar los procesos
11 'Variables que reciben valores ingresados por el usuario
12 a = 20
13 b = 50
14
15 suma = a + b
16
17 'Variables que reciben valores dentro del código
18 suma = a + b
19 resta = a - b
20 producto = a * b
21 div = a / b
22 modulo = a Mod b
23
24 'Bloque de código para mostrar resultados
25 Console.WriteLine("*** OPERACIONES MATEMATICAS ***\n")
26 Console.WriteLine("Suma: {0}", suma)
27 Console.WriteLine("Resta: ", resta)
28 Console.WriteLine("Producto: ", producto)
29 Console.WriteLine("División: ", div)
30 Console.WriteLine("Resto div: ", modulo)
31 Console.WriteLine("Operaciones realizadas con éxito")
  
```

```

C   PROGRAMA PARA CALCULAR FACTORIALES
   READ (5,100) N
   100 FORMAT (I5)
   NFAC=1
   500 CONTINUE
   IF (N) 1000, 1000, 600
   600 CONTINUE
   NFAC=NFAC*N
   N=N-1
   GOTO 500
1000 WRITE (6,110) NFAC
   110 FORMAT (I10)
   STOP
   END
  
```

```

program factorial(input,output);
var
  n:integer;
function fact(m:integer):integer;
begin
  if m>0 then
    fact:=m*fact(m-1)
  else
    fact:=1;
end;
  
```

4

PROGRAMACIÓN FUNCIONAL

- Es un paradigma de programación declarativa basado en la utilización de funciones aritméticas que no maneja datos mutables o de estado.
- La programación funcional tiene sus raíces en el cálculo lambda.
- Los lenguajes de programación puramente funcionales son utilizados principalmente en ambientes académicos y no en el desarrollo de software comercial.
- Algunos lenguajes de programación funcional son: Scheme, Erlang, Rust, Objective Caml y Haskell.

5

PROGRAMACIÓN FUNCIONAL

(sumaE xs) es la suma de los números de xs. Por ejemplo,
`sumaE [2,3,5] == 10`

```
sumaE :: [Int] -> Int
sumaE xs = sumaE' 0 xs
```

```
sumaE' :: Int -> [Int] -> Int
sumaE' v [] = v
sumaE' v (x:xs) = (sumaE' (v+x)) xs
```

6

PROGRAMACIÓN LÓGICA

- Es un tipo de paradigma de programación dentro de la programación declarativa.
- Se basa en el concepto de predicado, o relación entre elementos.
- Se utiliza en aplicaciones de inteligencia artificial o relacionadas, como son:
 - sistemas expertos,
 - demostración automática de teoremas, y
 - reconocimiento de lenguaje natural.
- Ejemplos de lenguajes que pertenecen a este paradigma de programación son: Prolog y LISP.

7

PROGRAMACIÓN LÓGICA

```

1 lisp
2 > Guardar valores como una lista de caracteres
3 > (define (SumarSiguiente V)
4   (cond ((null V) (progn (print "Suma=") 0))
5         (T (+ ( SumarSiguiente (cdr V)) (car V) ) ))
6 SUMARSIGUIENTE
7 > Crear vector de valores de entrada
8 (defun ObtenerEntrada (f c)
9   (cond ((eq c 0) nil)
10        (T (cons (read f) (ObtenerEntrada f (- c 1))))))
11 OBTENERENTRADA
12 > (defun Hazlo ()
13   (progn
14     (setq archivoent (open "lisp.data"))
15     (setq arreglo (ObtenerEntrada archivoent (read archivoent)))
16     (print arreglo)
17     (print (SumarSiguiente arreglo))))
18 HAZLO
19 >Hazlo
20
21 (1 2 3 4)
22 "Suma="
23 10
24 10

```

```

Arbol Gene Milton.pl [modified]
File Edit Browse Compile Prolog Poe Help
"scratch" [modified] Arbol_Gene_Milton.pl [modified]
hombre(cocinci). % Declarando Hombres
hombre(rogelio).
hombre(relix).
hombre(roge).
hombre(carlos).
hombre(milton).
hombre(julio).
hombre(daniel).
hombre(ranu).
hombre(chuco).

es_hombre(X) :- hombre(X). % Definiendo Hombre o Mujer
es_mujer(X) :- not(hombre(X)).

casados(rogelio,soco). % Declarando Parejas
casados(relix,kika).
es_matrimonio(X,Y) :- casados(X,Y) ; casados(Y,X).

es_hijo(X,Y) :- progenitor(Y,X) , es_hombre(X). % Definiendo algunos Parentescos
es_hijo(X,Y) :- progenitor(Z,X) , es_matrimonio(Z,Y) , es_hombre(X).
es_hija(X,Y) :- progenitor(Y,X) , es_mujer(X).
es_hija(X,Y) :- progenitor(Z,X) , es_matrimonio(Z,Y) , es_mujer(X).

son_hermanos(X,Y) :- progenitor(Z,X) , progenitor(Z,Y).
son_hermanos(X,Y) :- progenitor(Z,X) , progenitor(Z,Y) , es_mujer(X) , es_mujer(Y).

es_padre(X,Y) :- progenitor(X,Y) , es_hombre(X).
es_madre(X,Y) :- es_matrimonio(X,Z) , progenitor(Z,Y) , es_mujer(X).

```

8

PARADIGMA DE PROGRAMACIÓN ORIENTADA A OBJETOS

9

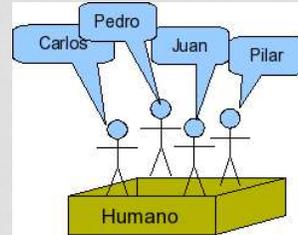
PROGRAMACIÓN ORIENTADA A OBJETOS

- Facilita la **creación** de software de calidad pues sus características facilitan:
 - La mantenibilidad
 - La extensibilidad
 - La reutilización del software
 - La usabilidad del software generado bajo este paradigma.
- La POO se basa en la **idea natural** de un mundo lleno de **objetos** y que la resolución de problemas se realiza mediante el modelo de objetos.
- Los conceptos de clase y objeto son los más importantes de la POO.

10

PROGRAMACIÓN ORIENTADA A OBJETOS

- **Objeto** es cualquier cosa tangible o intangible que se pueda imaginar definida mediante sus atributos y las operaciones que permiten modificar dichos atributos.
- Una **clase** es una plantilla que permite definir un conjunto de objetos.



11

PROGRAMACIÓN ORIENTADA A OBJETOS

- Es un paradigma de programación que usa los objetos en sus interacciones para diseñar aplicaciones y programas.
- Se fundamenta en los siguientes principios:
 - Abstracción
 - Encapsulamiento
 - Modularidad
 - Herencia
 - Paso de mensajes
 - Polimorfismo
- Se popularizó a principios de la década de los años 1990.

12

CONCEPTOS DE LA POO

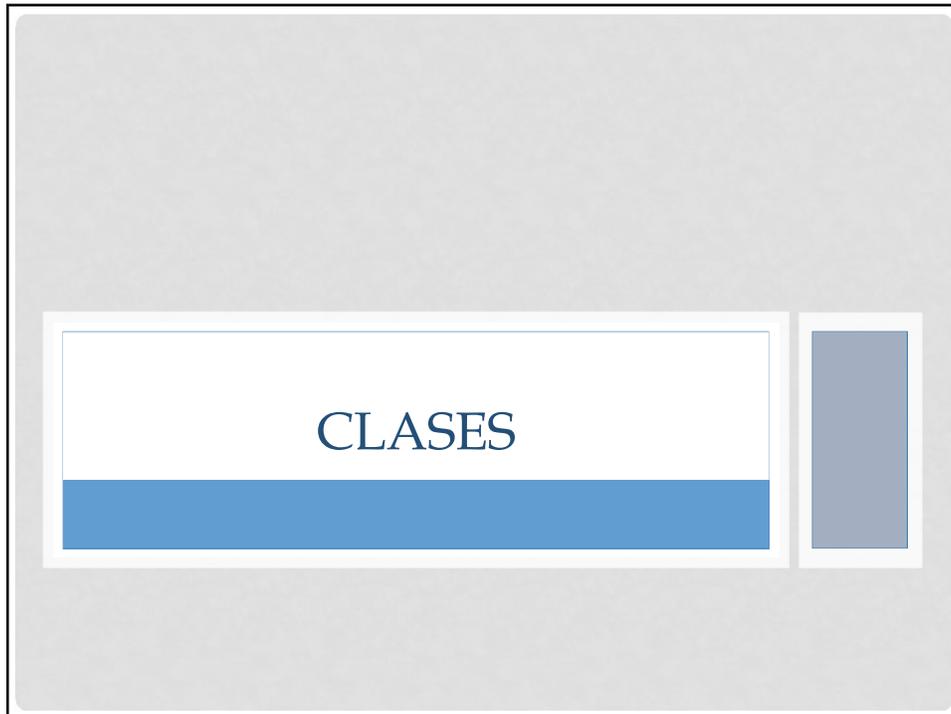
- La POO trata de aproximarse al modo de **actuar y pensar del hombre** y no al de la máquina.
- Conceptos fundamentales que sustentan la POO:
 - Clase
 - Objeto
 - Instancia
 - Atributos
 - Métodos

13

PROGRAMACIÓN ORIENTADA A OBJETOS

- Los objetos se pueden **Organizar** de acuerdo a sus características comunes:
 - **Mamíferos** : Perro, Ciervo
 - **Teléfonos** : Fijo, Celular
 - **Deportes** : Fútbol, Tenis
 - **Vehículos** : Automóvil, Camión

14



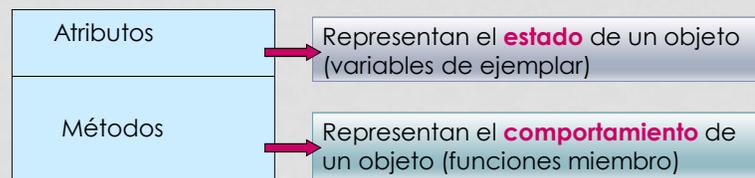
CLASES

- Las clases **son abstracciones** que representan a un conjunto de objetos con un:
 - Comportamiento e
 - Interfaz común
- Es la **implementación** de **un tipo de dato** (considerando los objetos como instancias de las clases)

16

CLASES

- Permiten definir y representar colecciones de objetos
- Proveen un **modelo** para la creación de objetos
- Los elementos que componen la clase son



17

CLASE

- Una clase es como un tipo de dato creado por el usuario, que posee dos categorías de miembros:
 - **Atributos** (Datos) ⇒ Estado
 - **Métodos** (Algoritmos) ⇒ Comportamiento

18

CLASE

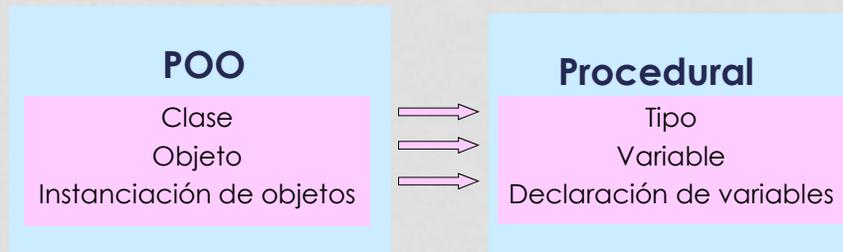
- En general, es posible crear una clase a partir de cualquier objeto que esté a nuestro alrededor.
- Por ejemplo:
 - Persona
 - Automóvil
 - Mascota

19

OBJETOS

OBJETOS

- ✚ En la **POO** se dice que un objeto: "integra datos y algoritmos"
- ✚ En la **programación estructurada**, variables y funciones están separadas



21

OBJETOS

- ✚ Cada objeto **es responsable** de inicializarse y destruirse en forma correcta
- ✚ Un **objeto** consta de:
 - [Tiempo de vida](#)
 - [Estado](#)
 - [Comportamiento](#)



22

TIEMPO DE VIDA DE UN OBJETO

- ✚ La duración de un objeto en un programa siempre está limitada en el tiempo
- ✚ La mayoría de los objetos sólo existen durante una parte de la ejecución del programa
- ✚ Los objetos son creados mediante un mecanismo denominado **instanciación**
- ✚ Los objetos dejan de existir cuando son **destruidos**

23

ESTADO DE UN OBJETO

- ✚ Queda definido por sus **atributos**
- ✚ Con él se definen las **propiedades del objeto**, y el estado en que se encuentra en un momento determinado de su existencia

24

COMPORTAMIENTO DE UN OBJETO

- ✦ Queda definido por los **métodos**
- ✦ Los prototipos de los **métodos** definidos en la interfaz de una clase permiten a otros objetos, que forman parte de la aplicación, interactuar con los objetos de esa clase

25

INSTANCIA

26

INSTANCIAS-OBJETOS

- Una **Instancia** es una ocurrencia de la clase
- Al momento de crear un objeto se produce la instanciación
- Un **Objeto** es una instancia de una Clase específica

27

INSTANCIAS-OBJETOS

EJEMPLO:

Suponer que existe la clase **Vehículo**

- El auto con placas WF-4631 es una instancia de la clase **Vehículo**, o sea, un objeto de esa clase
-
- La camioneta con placas ZT-9283 es otra instancia de la clase **Vehículo**

28

ATRIBUTOS

- Son los datos que caracterizan a los objetos de una clase y determinan el estado de un objeto

- Marca
- Año
- Color
- Placas, etc.

29

MÉTODOS

- Representan todas aquellas **acciones** que se pueden realizar **sobre un objeto** de cierta clase
- En la implementación, estos métodos son segmentos de código en la forma de funciones
- La clase **Vehículo** puede incluir los métodos:
 - Encender
 - Acelerar
 - Virar
 - Frenar

30

EJEMPLO DE INSTANCIA DE OBJETOS

- **Clase:** Cuenta corriente
 - Atributos:
 - Número
 - Nombre
 - Saldo
 - Métodos:
 - Depositar
 - Girar
 - Consultar saldo

31

EJEMPLO DE INSTANCIA DE OBJETOS

- **Clase:** Cuenta corriente
- **Instanciación:** Cuenta Corriente A, B

Objeto: **A**Objeto: **B**

32



MÉTODOS

- Un método es una función miembro de una clase.
- Establece el comportamiento del objeto.
- Opera directamente sobre el objeto que lo invocó.
- Recibe, como **parámetro implícito**, el objeto que lo invocó.

- Tipos de métodos
 - Constructores
 - Accesores
 - Otros

34

MÉTODOS

- Si el método requiere otros objetos de la clase, éstos deberán ser pasados como parámetros explícitos y el método sólo podrá acceder en forma indirecta a estos objetos
- Ejemplos:
 - Depositar
 - Girar
 - Consultar

35

MENSAJES Y MÉTODOS

- Un objeto (agente emisor) envía un mensaje a otro objeto (agente receptor)
- El mensaje tiene codificada la petición de una acción
- El mensaje incluye la información (argumentos) necesaria para satisfacer la petición
- Si el receptor acepta el mensaje, acepta la responsabilidad de ejecutar la acción indicada
- En respuesta a un mensaje, el receptor ejecuta un método para satisfacer la petición

36

CLASES Y EJEMPLARES

- Todos los objetos son ejemplares de una clase
- La clase del receptor determina el método que se activa como respuesta a un mensaje
- Todos los objetos de una clase usan el mismo método en respuesta a mensajes similares

37

CLASES Y MÉTODOS

- Los objetos son ejemplos de TAD's
- Un TAD tiene dos caras: una exterior, la que ve el usuario, y una interior, la que sólo ve el programador
- El usuario ve nada más que un conjunto de operaciones que definen el comportamiento de la abstracción
- El programador ve las variables de datos que se usan para mantener el estado interno del objeto
- Un ejemplar es un representante de una clase

38

CLASES Y MÉTODOS

- El estado lo determinan las variables de ejemplar
- El comportamiento lo determinan los métodos
- Desde el exterior, los clientes sólo pueden ver el comportamiento de los objetos
- Desde el interior, los métodos proporcionan el comportamiento apropiado mediante las modificaciones del estado

39

CLASES Y MÉTODOS

- La interfaz describe la forma en que un objeto se conecta con el mundo
- La implementación describe cómo se logra la responsabilidad prometida en la interfaz
- Una clase se puede concebir como un registro con dos variedades de campos: datos y procedimientos
- Los datos constituyen las variables de ejemplar
- Los procedimientos constituyen los métodos

40

DUDAS O COMENTARIOS???