

## **TEMA 2**

## **TIPOS ABSTRACTO DE DATOS**

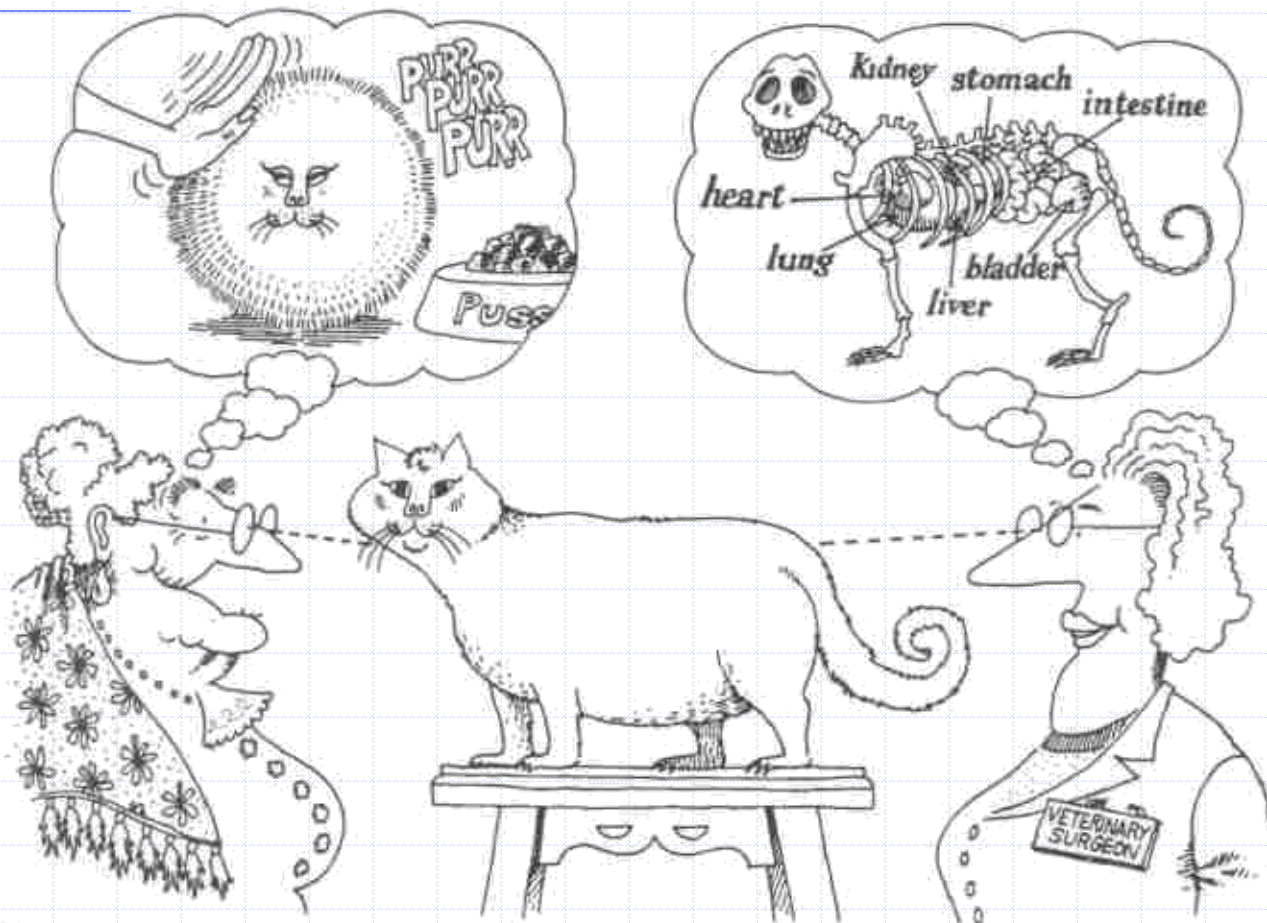
# INTRODUCCION

- ◆ Abstracción
- ◆ Estructura de Datos
- ◆ Tipos de Datos Abstractos

# 1.1 Abstraccion

- ◆ Abstracción: Es la representación de las características esenciales de un objeto o entidad.
- ◆ Abstraer: Dar nombre a las cosas.
- ◆ Referenciar: Hacer uso del nombre.

# Abstracción



Abstraction focuses upon the essential characteristics of some object, relative to the perspective of the viewer.

# 1.1 Abstracción

## ¿Por qué es importante?

- ◆ El proceso de abstracción, debe convertirse en una habilidad para quien estudie una carrera relacionada con la computación (informática, sistemas, redes). La capacidad de modelar una realidad por medio de herramientas computacionales requiere necesariamente de hacer continuas abstracciones, por lo que es vital utilizar metodologías que desarrollen esta habilidad.

# 1.1 Abstracción

## ¿Qué es la abstracción de datos?

- ◆ **La abstracción de datos** es una técnica o metodología que permite diseñar estructuras de datos.
- ◆ Consiste básicamente en *representar* bajo ciertos lineamientos de formato las *características esenciales* de una estructura de datos.
- ◆ Este proceso de diseño se *olvida de los detalles específicos de implementación* de los datos.

# 1.2 Estructura de Datos

## ¿Qué es una estructura de datos?

- ◆ Cualquier colección o grupo de datos organizados de tal forma que tengan asociados un conjunto de operaciones para poder manipularlos, se dice que conforma una estructura de datos.

# 1.3 Tipo de Dato Abstracto (TDA)

- ◆ Es la representación de una entidad u objeto para facilitar su programación. Se compone de:
  - **Estructura de datos:** Es la estructura de programación que se selecciona para representar las características de la entidad modelada
  - **Funciones de Abstracción:** Son funciones que permiten hacer uso de la estructura de datos, y que esconden los detalles de dicha estructura, permitiendo un mayor nivel de abstracción.



## 1.3.1 Especificación lógica de un TDA

- ◆ Se plasma la abstracción realizada al diseñar una estructura de datos, esto pasa a ser el mapa o plano con el cual se construirá la estructura de datos y se definirán claramente las reglas en las que podrá usarse el TDA. La especificación lógica de un TDA consiste de los siguientes cuatro puntos:

## 1.3.2 Especificación lógica de un TDA

### ◆ I. Elementos que conformarán la estructura de datos.

Es el tipo de los datos que se guardará en la estructura.

Ejemplo:

números enteros, caracteres, fechas, registros con los datos de un empleado, etcétera.

## 1.3.2 Especificación lógica de un TDA

### ◆ 2. Tipo de organización en que se guardarán los elementos.

Existen cuatro tipos de organización para los datos en la estructura.

- **Lineal**: Si hay una relación de uno a uno entre los elementos.
- **Jerárquica**: Si hay una relación de uno a muchos entre los elementos.
- **Red**: Si hay una relación de muchos a muchos entre los elementos.
- **Sin relación**: Si no hay relaciones entre los elementos.

## 1.3.2 Especificación lógica de un TDA

### ◆ 3. Dominio de la estructura.

Este punto es opcional, y en él se describirá la capacidad de la estructura en cuanto al rango posible de datos por guardar.

## 1.3.2 Especificación lógica de un TDA

### ◆ 4. Descripción de las operaciones de la estructura.

Cada operación que está relacionada con la estructura debe describirse como:

- Nombre de la operación.
- Descripción breve de su utilidad.
- Datos de entrada a la operación.
- Datos que genera como salida la operación.
- Precondición: Condición que deberá cumplirse antes de utilizar la operación para que se realice sin problemas.
- Poscondición: Condición en que queda el TDA después de ejecutar la operación.

# Ejemplo:

## ◆ *Especificación lógica del TDA: Cadena*

- **Elementos:** todos los caracteres alfabéticos (letras mayúsculas y minúsculas), caracteres numéricos y caracteres especiales.
- **Estructura:** hay una relación lineal entre los caracteres
- **Dominio:** existen entre 0 y 80 caracteres en cada valor del TDA CADENA. El dominio serán todas aquellas secuencias de caracteres que cumplan con las reglas.

# Ejemplo:

## ◆ Operaciones:

### **Borrainicio**

- ◆ Utilidad: Sirve para eliminar el primer carácter de una cadena.
- ◆ Entrada: Cadena S sobre la que se desea eliminar el primer carácter.
- ◆ Salida: El carácter más a la izquierda de la cadena S y la cadena S modificada.
- ◆ Precondición: La cantidad de caracteres es mayor que cero.
- ◆ Poscondición: La cadena S tiene todos los caracteres, menos el primero.

# Ejemplo:

## **Agregafinal**

- ♦ Utilidad: Sirve para agregar un carácter al final de una cadena.
- ♦ Entrada: Cadena S y el carácter L, que se añadirá a la cadena S.
- ♦ Salida: Cadena S modificada.
- ♦ Precondición: La cantidad de caracteres en S es menor que 80.
- ♦ Poscondición: La cadena S tiene el carácter L que queda al extremo derecho de la cadena.



# Ejemplo ....

## Vacía

- ♦ Utilidad: Sirve para verificar si una cadena está vacía o no.
- ♦ Entrada: Cadena S que se verificará.
- ♦ Salida: VERDADERO si la cadena S no tiene caracteres, FALSO en caso contrario.
- ♦ Precondición: Ninguna
- ♦ Poscondición: Ninguna (pues la cadena S no se modifica).

# Ejemplo ....

## Llena

- ♦ Utilidad: Sirve para verificar si una cadena está llena o no.
- ♦ Entrada: .Cadena S que será verificada.
- ♦ Salida: VERDADERO si la cadena S contiene ya 80 caracteres, FALSO en caso contrario
- ♦ Precondición: Ninguna
- ♦ Poscondición: Ninguna (pues la cadena S no se modifica).

# Ejemplo ....

## Invierte

- ♦ Utilidad: Sirve para invertir el orden de los caracteres en una cadena.
- ♦ Entrada: Cadena S a la que se desea invertir el orden de los caracteres.
- ♦ Salida: Cadena S modificada.
- ♦ Precondición: Ninguna
- ♦ Poscondición: La secuencia de caracteres en la cadena S se invierte, de forma que el primer carácter toma el lugar del último, el segundo el del penúltimo y así sucesivamente.

## 1.3.3 Niveles de abstracción de datos

◆ En la abstracción de datos se pueden definir tres niveles de trabajo:

1. El nivel **lógico o abstracto** se define la estructura de datos y las operaciones relacionadas con ella. La descripción es independiente del lenguaje de programación en el se usará la estructura.

## 1.3.3 Niveles de abstracción de datos

- ◆ 2. El nivel **físico** o de implementación. En este nivel se decide el lenguaje de programación para la implementación, los tipos de datos ya definidos servirán para representarla y se implementa como un módulo (o método) a cada una de las operaciones del TDA. Este nivel toma el diseño del nivel lógico
- ◆ 3. En el nivel **aplicación** o de uso el programador usará el TDA para resolver determinada aplicación. El uso del TDA se limita a llamar las operaciones sobre la estructura que se requiera cuidando siempre de cumplir con las reglas de cada operación especificadas en el nivel lógico.

## 1.3.4 La independencia de datos y el ocultamiento de información

- ◆ Quien implementa el TDA no debe estar influenciado por la aplicación que tendrá la estructura, y quien use la estructura no tiene porqué saber cómo se implementaron sus operaciones.
- ◆ Entonces la forma en que se almacenan los datos en la estructura es independiente de su aplicación y que para el usuario programador permanece oculto cómo se implementaron las operaciones del TDA.

## 1.3.5 ¿Cómo distinguir los niveles de abstracción?

Una analogía podría hacerse al comparar este proceso con el que se realiza al construir una casa. Primero se trabaja en el nivel lógico al pedirle a un arquitecto que diseñe el plano. Una vez aprobado el diseño, se procede en el nivel físico, que correspondería a la construcción. Los albañiles seguirán paso a paso las especificaciones dadas en el plano. Una vez que la construcción esté terminada, llegará su dueño para habitarla dándole el uso correspondiente a cada espacio.

## 1.3.6 ¿Qué ventajas ofrece utilizar la técnica de abstracción de datos?

- ◆ La técnica obliga a diseñar modularmente y, como consecuencia se tiene una implementación más clara, documentada y es fácil darle mantenimiento. Adicionalmente, gracias a la independencia de datos y al ocultamiento de información se pueden crear paquetes como unidades de software reutilizable con lo que se obtienen estructuras de datos genéricas.