

# Estructuras de Datos

## Tipos de Datos Abstractos

**Adriana Collaguazo Jaramillo, Mg.**

# Recordatorio



# Contenidos (resumen)

## Primer parcial

- Tipos de Datos y Tipos de Datos Abstractos (TDAs)
- Listas
- Pilas
- Colas
- Conjuntos y mapas

## Segundo parcial

- Grafos
- Árboles

# Contenidos (resumen)

## Primer parcial

- ***Tipos de Datos y Tipos de Datos Abstractos (TDAs)***
- Listas
- Pilas
- Colas
- Conjuntos y mapas

## Segundo parcial

- Grafos
- Árboles

# Tipos de Datos

# TIPOS DE DATOS

- Los datos se clasifican en TIPOS
  - Diferentes dominios existentes

- Edad
- Año de Nacimiento
- Número de multas

Dominio **numérico**

- Nombre
- Dirección
- Cédula

Dominio **texto**

- Operaciones asociadas para dicho dominio

# Datos Primitivos

- Los podemos distinguir fácilmente, están en el diario vivir:
  - El Sr. Vera de 63 años tiene cedula No. 0908815533, y paga \$120 de impuestos
- Permiten representar información numérica, caracteres, etc.

NOMBRE	CONJUNTO DE VALORES	OPERACIONES
Enteros	Negativos y positivos sin decimal	Suma, resta, multiplicación, división, residuo
Reales	Negativos y positivos, con decimal	Suma, resta, multiplicación, división, residuo
Lógicos	Verdadero (True) => 1 Falso (False)=> 0	And, Or, Not
Caracteres	Letras, números, especiales, juntos forman una cadena	Concatenar (suma de cadenas), buscar, subcadena, etc.

# Datos Compuestos

- En ocasiones se necesitan tipos de datos más complejos y **estructurados**
  - Variables que almacenen más de un valor
  - Variables que representen información de la vida real
  - Estarán formados a partir de tipos de datos simples
- En Java, tenemos:

TIPO	FORMATO DECLARACION
ARREGLOS	<code>int arreglo[];</code>
CLASES	<code>public class ClaseEjemplo {  }</code>
COLECCIONES	Set, List, Map, Queue

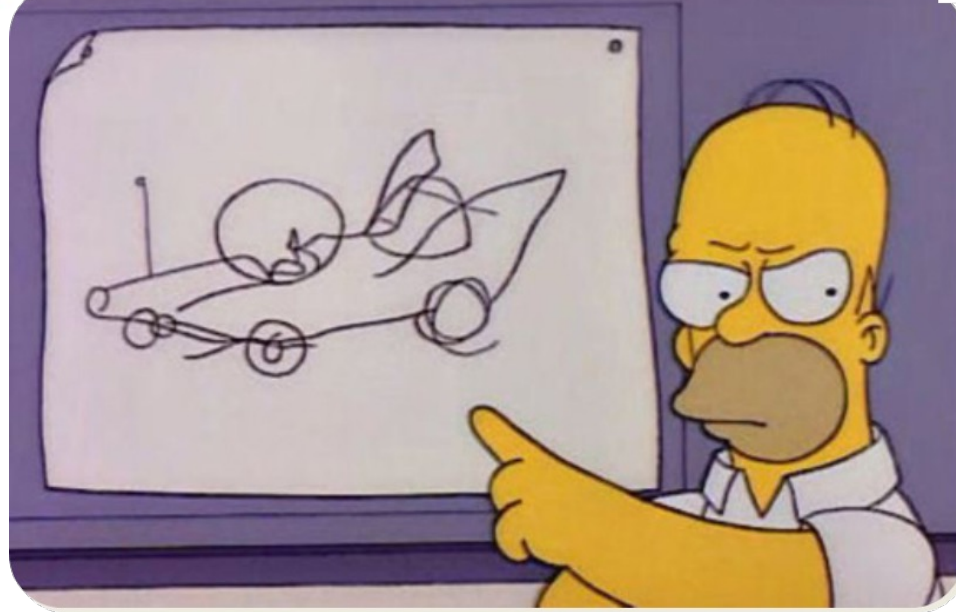


# **Tipos de Datos Abstractos (TDAs)**

# Abstracción

- Visión simplificada de una realidad en la que sólo consideramos determinados aspectos esenciales.
- Consiste en enfocarse en lo esencial.

# ¿Qué significa abstracción?



Énfasis en el  
¿qué hace? mas  
que en el ¿cómo  
lo hace?

Homero Simpson construyendo el auto de sus sueños

# En Programación

## **Abstracción de Control**

- Se refiere a la utilización de procedimientos o funciones sin preocuparse de cómo se implementan.
- Basta con saber qué hace el procedimiento.
- Ocultación de los detalles de la implementación (variables, secuencias, etc).

## **Abstracción de Datos**

- Permite definir nuevos tipos de datos.
- Permite diseñar programas más cortos, legibles y flexibles.
- Tipo de Dato Abstracto (TDA)

# Tipo de Dato Abstracto (TDA)

- Un TDA es un modelo matemático de una estructura de datos.
- Un TDA trata de representar entidades del mundo real especificando el **QUÉ** y no el **CÓMO**.
- Se compone de:
  - **Comportamiento u operaciones**
    - Los TDAs existen para proveer operaciones
    - Ejemplo: Un carro, es útil porque se lo puede **manejar**
    - **manejar** es un comportamiento u operación del TDA Carro
  - **Propiedades o Atributos**
    - Se refiere al funcionamiento interno del TDA
    - Un TDA correctamente creado mantiene sus propiedades OCULTAS
    - Ejemplo: No nos interesa cómo funciona un carro; solo nos interesa que funcione

# Implementación de un TDA

- En JAVA, los TDAs pueden ser implementados mediante:
  - Clases
  - Clases Abstractas
  - Interfaces

# Ejemplo TDA List

Representa una colección de elementos de cualquier tipo

A rectangular box with a blue border containing the text "List" and "<interface>" stacked vertically.

List  
<interface>

Permite añadir, eliminar y recuperar elementos

Permite elementos duplicados (un mismo elemento puede ser añadido varias veces)

No admite valores nulos

Las operaciones de añadir y eliminar devuelven valores booleanos de éxito

# Ejemplo TDA List

Representa una colección de elementos de cualquier tipo

List  
<interface>

## Method and Description

**add(E e)**

Appends the specified element to the end of this list (optional operation).

**add(int index, E element)**

Inserts the specified element at the specified position in this list (optional operation).

**addAll(Collection<? extends E> c)**

Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator (optional operation).

**addAll(int index, Collection<? extends E> c)**

Inserts all of the elements in the specified collection into this list at the specified position (optional operation).

**clear()**

Removes all of the elements from this list (optional operation).

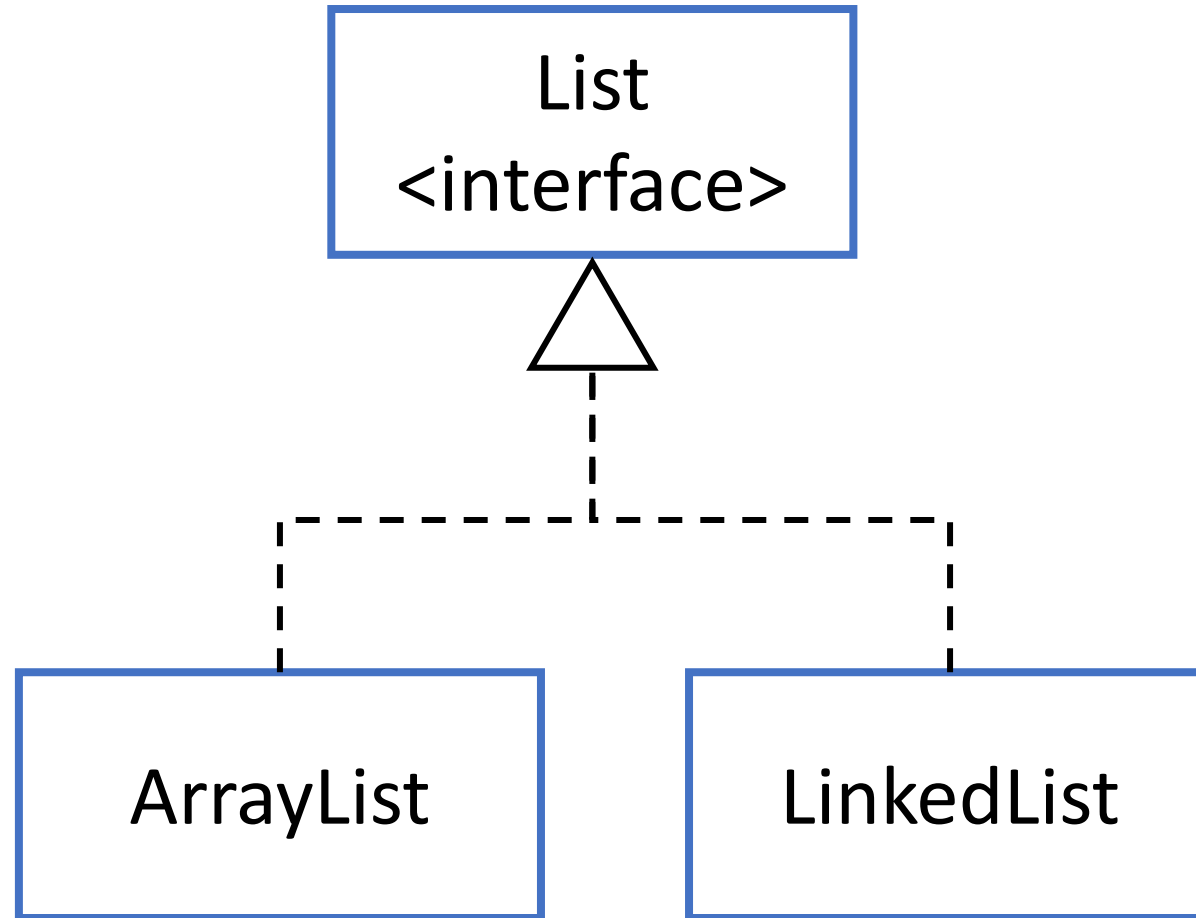
**contains(Object o)**

Returns **true** if this list contains the specified element.



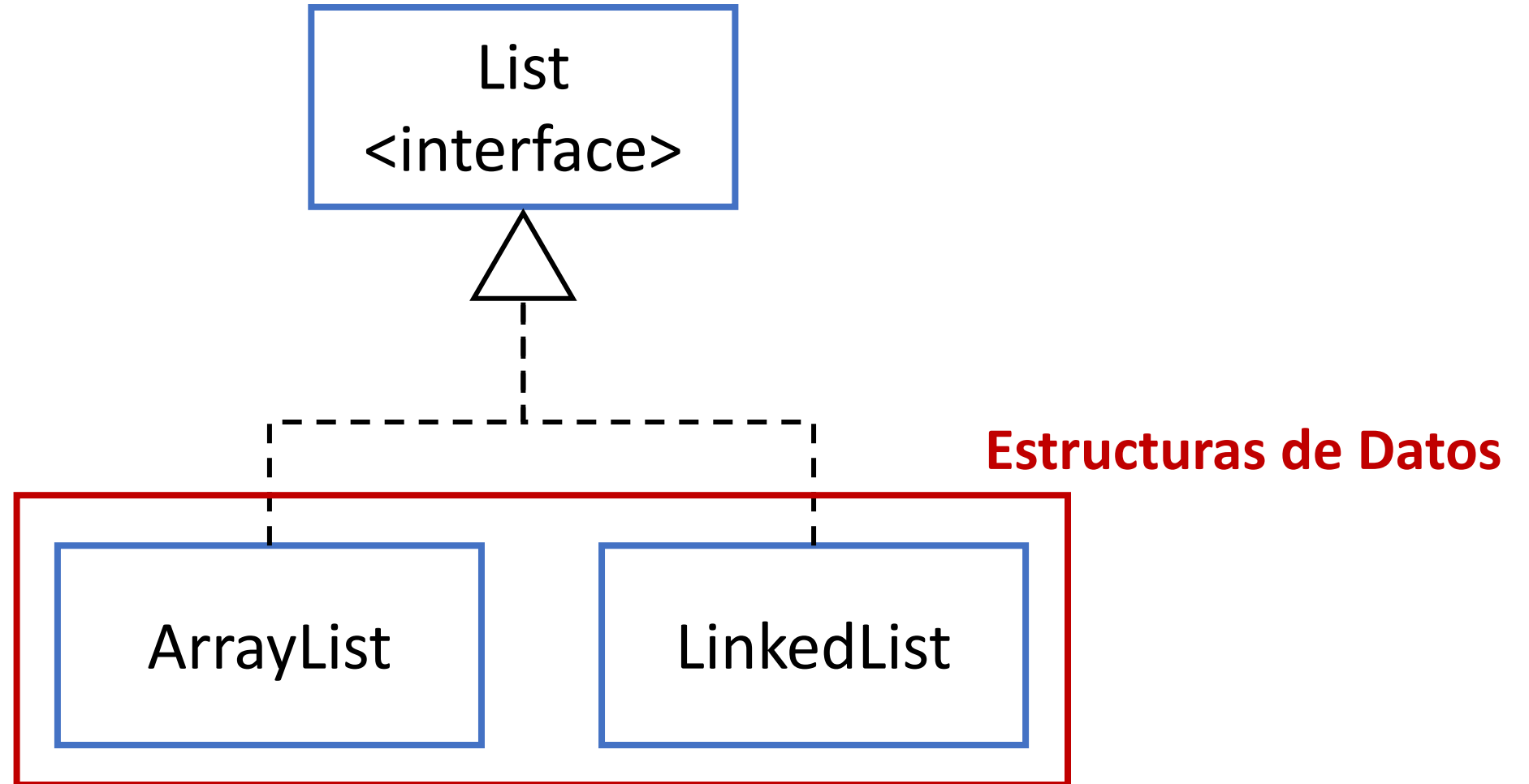
# Ejemplo TDA List

Representa una colección de elementos de cualquier tipo



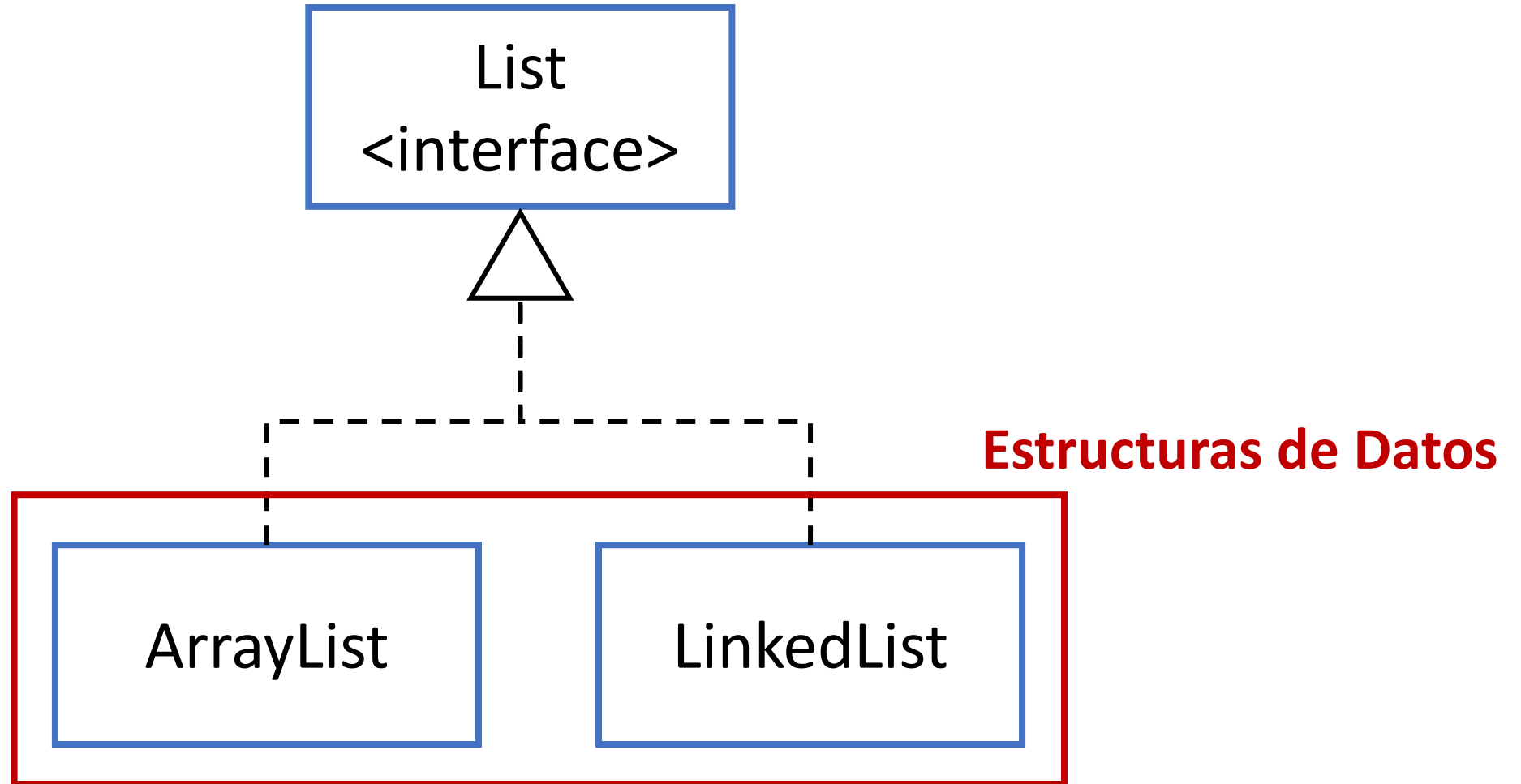
# Ejemplo TDA List

Representa una colección de elementos de cualquier tipo



# Ejemplo TDA List

Representa una colección de elementos de cualquier tipo



Se usan de manera similar, lo que varía es la implementación interna

# Ejemplo TDA List

Para un programador que usa las estructuras:

Tanto el `ArrayList` como la `LinkedList` son listas

El usuario no se entera de los detalles de implementación

Para quien implementa las estructuras:

Éstas son representaciones **concretas** del TDA `List`

Debe tomar decisiones de cómo lograr el comportamiento definido del TDA

En este curso, usted implementará varios TDAs

# Antes de todo...

Necesitamos conocer algunos conceptos fundamentales:

- Java Generics
- Notación asintótica
- Recursividad