



FUNDACIÓN UNIVERSITARIA DE POPAYÁN

Educación de **calidad** con
responsabilidad **social**



FUNDACIÓN
UNIVERSITARIA
DE POPAYÁN

INGENIERIA DE SISTEMAS

Estructura de Datos

Adrian Danilo Astudillo

Mail: adrian.astudillo@docente.fup.edu.co



FUNDACIÓN
UNIVERSITARIA
DE POPAYÁN

Objetivo

Desarrollar programas informáticos que utilizan estructuras de datos para la solución de problemas propios de la ingeniería, integrando técnicas y herramientas de programación orientada a objetos

CONTENIDO

- ✓ Introducción a Listas
- ✓ Sintaxis Listas
 - Declaración
 - Definición e inicialización de variables
 - Operadores
 - Estructuras de Control
 - Clases y Objetos

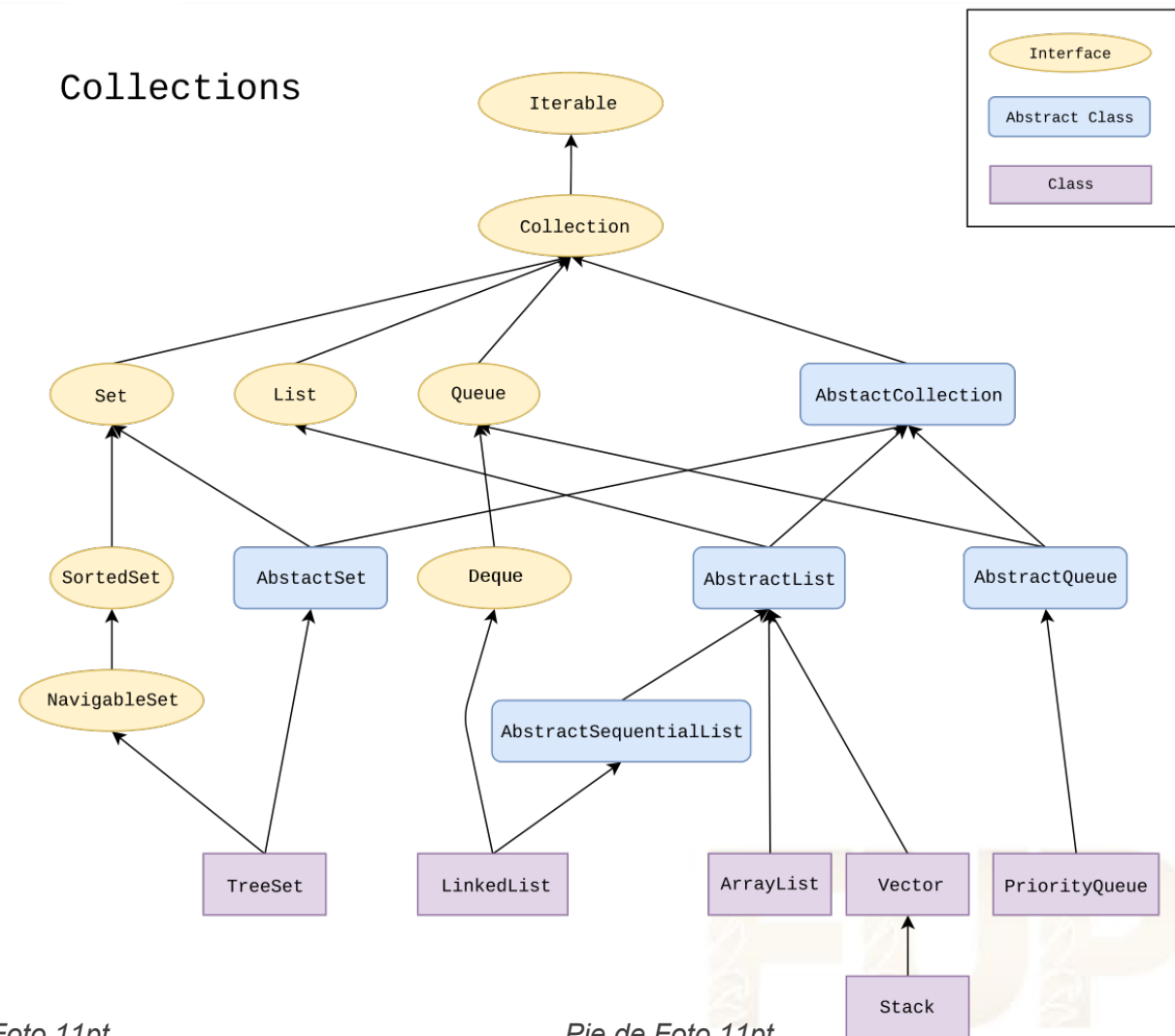


INTRODUCCIÓN

Definición

Las listas son colecciones ordenadas de objetos. Son similares a las secuencias en matemáticas en ese sentido. Sin embargo, son diferentes a los conjuntos que no tienen un cierto orden.

Collections



EJEMPLO

Usaré ArrayList en los primeros ejemplos, porque es el tipo de lista más utilizado.

ArrayList es básicamente un arreglo de tamaño variable. Casi siempre, vas a querer usar ArrayList en lugar de arreglos regulares, ya que brindan muchos métodos útiles.

La única ventaja de un arreglo solía ser su tamaño fijo (al no asignar más espacio del que necesita). Pero las listas también admiten tamaños fijos ahora.



EJEMPLO

```
import java.util.ArrayList;
import java.util.List;

public class CreateArrayList {
    public static void main(String[] args) {
        ArrayList<Integer> list0 = new ArrayList<>();

        // Makes use of polymorphism
        List list = new ArrayList<Integer>();

        // Local variable with "var" keyword, Java 10
        var list2 = new ArrayList<Integer>();
    }
}
```

EJEMPLO

```
import java.util.ArrayList;
import java.util.List;

public class CreateArrayList {
    public static void main(String[] args) {
        ArrayList<Integer> list0 = new ArrayList<>();

        // Makes use of polymorphism
        List list = new ArrayList<Integer>();

        // Local variable with "var" keyword, Java 10
        var list2 = new ArrayList<Integer>();
    }
}
```

En los corchetes angulares (<>) especificamos el tipo de objetos que vamos a almacenar.

Tenga en cuenta que el tipo entre paréntesis debe ser un tipo de objeto y no un tipo primitivo. Por lo tanto, tenemos que usar envoltorios de objetos, clase Integer en lugar de int, Double en lugar de double, y así sucesivamente.

EJEMPLO

Hay muchas formas de crear una `ArrayList`, pero presenté tres formas comunes en el fragmento anterior.

La primera forma es creando el objeto a partir de la clase `ArrayList` concreta, especificando `ArrayList` en el lado izquierdo de la asignación.

```
ArrayList<Integer> list0 = new ArrayList<>();
```



EJEMPLO

El segundo fragmento de código hace uso del polimorfismo mediante el uso de la lista en el lado izquierdo. Esto hace que la asignación se acople libremente con la clase `ArrayList` y nos permite asignar otros tipos de listas y cambiar fácilmente a una implementación de Lista diferente.

```
// Makes use of polymorphism  
List list = new ArrayList<Integer>();
```

EJEMPLO

La tercera forma es la forma Java 10 de crear variables locales haciendo uso de la palabra clave var. El compilador interpreta el tipo de variable comprobando el lado derecho.

```
// Local variable with "var" keyword, Java 10  
var list2 = new ArrayList<Integer>();
```

EJEMPLO

La tercera forma es la forma Java 10 de crear variables locales haciendo uso de la palabra clave `var`. El compilador interpreta el tipo de variable comprobando el lado derecho.

```
// Local variable with "var" keyword, Java 10  
var list2 = new ArrayList<Integer>();
```

Cómo agregar y actualizar elementos de lista en Java

Para agregar elementos a la lista podemos usar el método `add`. También podemos especificar el índice del nuevo elemento, pero ten cuidado al hacerlo, ya que puedes generar una excepción `IndexOutOfBoundsException`.



Agregar

```
import java.util.ArrayList;

public class AddElement {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("hello");
        list.add(1, "world");
        System.out.println(list);
    }
}
```


Resultado y Actualizar

```
[hello, world]
```

Podemos usar el método set para actualizar un elemento.

```
list.set(1, "from the otherside");  
System.out.println(list);
```

Resultado:

```
[hello, world]  
[hello, from the otherside]
```

Recorrer

Java – Ejercicios

- ✓ Elaborar un algoritmo que permita leer 20 números enteros en un arrayList e imprimirlos.
- ✓ Elaborar un algoritmo que permita leer 15 nombres de personas y los imprima



FUNDACIÓN
UNIVERSITARIA
DE POPAYÁN

Gracias