UNIVERSIDAD TECNOLÓGICA DEL PERÚ

Facultad de Ingeniería

Programacion Orientada a Objetos

Sesión 11:

Procesamiento de colecciones de objetos

Recordando:

¿Que vimos la clase pasada?

Universidad Tecnológica del Perú

Logro de aprendizaje

Al finalizar la unidad, el estudiante soluciona problemas aplicando colecciones usando Java.

Universidad Tecnológica del Perú

Utilidad

- Permite al estudiante tomar decisiones informadas sobre qué tipo de colección usar en función de los requisitos.
- Les permite búsqueda, ordenación y filtrado en una variedad de estructuras de datos, manipular datos de manera efectiva utilizando las colecciones.

Agenda

- Procesamiento de colecciones de objetos

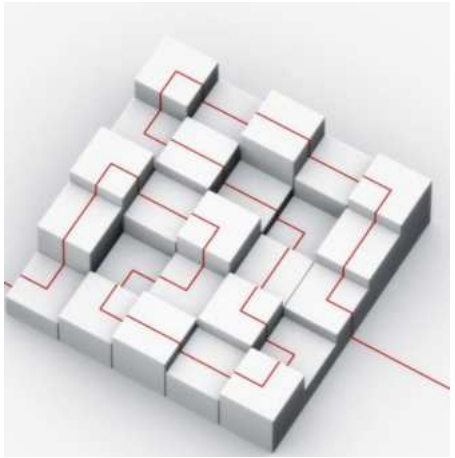
Universidad Tecnológica del Perú

Procesamiento de colecciones de objetos

- El procesamiento de colecciones de objetos en Java se refiere a la manipulación y el trabajo con conjuntos de objetos (o elementos) que se almacenan en estructuras de datos de colección, como Listas, Conjuntos, Mapas, Arrays, entre otros. Java proporciona una amplia variedad de clases y interfaces en su biblioteca estándar para manejar colecciones de objetos de manera eficiente.
- Algunos de los conceptos clave relacionados con el procesamiento de colecciones de objetos en Java incluyen:

Procesamiento de colecciones de objetos

- ✓ **Colecciones:**
  Las colecciones son estructuras de datos que permiten almacenar y organizar múltiples objetos. Algunos ejemplos de colecciones en Java son ArrayList, LinkedList, HashSet, TreeMap, etc.
- ✓ **Interfaces de colección:**
  Java proporciona interfaces como List, Set y Map que representan diferentes tipos de colecciones. Estas interfaces definen métodos comunes que se pueden utilizar para trabajar con elementos dentro de las colecciones.
- ✓ **Iteración:**
  Para procesar los elementos dentro de una colección, se utilizan bucles o iteradores. Los bucles for-each (también conocidos como bucles mejorados) son especialmente útiles para iterar sobre colecciones en Java.



https://www.youtube.com/watch?v=D0VH50zFVIA

Universidad Tecnológica del Perú

Ejemplo de procesamiento de una lista de elementos:

```
List<String> nombres = new ArrayList<>();
nombres.add("Juan");
nombres.add("María");
nombres.add("Pedro");
for (String nombre : nombres) {
System.out.println(nombre);
}
```

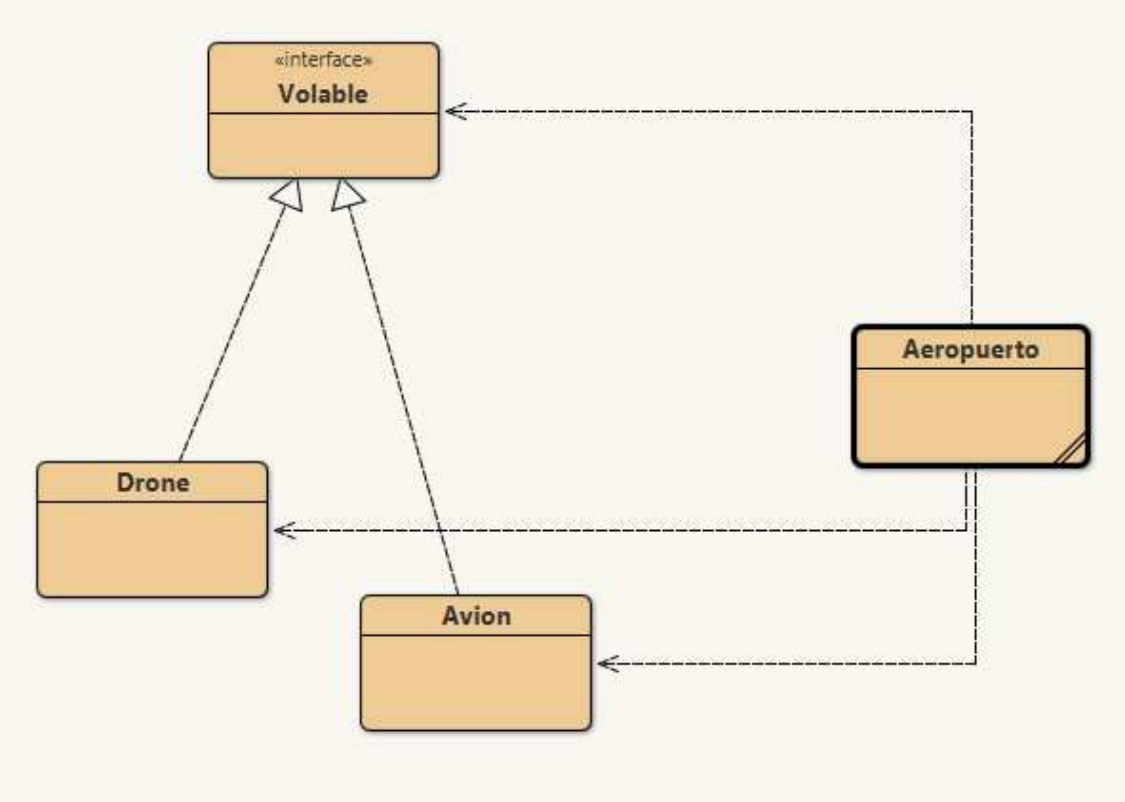
Métodos de colección:
Las colecciones en Java ofrecen una variedad de métodos para agregar, eliminar, buscar y modificar elementos.
Por ejemplo, add(), remove(), contains(), size(), entre otros.

Iteradores:
Los iteradores permiten recorrer una colección uno a uno.
Puedes obtener un iterador para una colección llamando al método iterator() en la colección.

Clases utilitarias:
Java también proporciona clases utilitarias en el paquete java.util para realizar operaciones comunes en colecciones, como Collections para ordenar y buscar, y Arrays para trabajar con arrays.

Iteradores

```
Iterator<String> iterator =
nombres.iterator();
while (iterator.hasNext()) {
    String nombre = iterator.next();
    System.out.println(nombre);
}
```



Ejercicio 1: Aeropuerto

```
// Interfaz Volable
public interface Volable {
    void despegar();
    void volar();
    void aterrizar();
}
```

Ejercicio 1: Aeropuerto

```
// Clase Avion que implementa Volable
public class Avion implements Volable {
    private String modelo;
    public Avion(String modelo) {
        this.modelo = modelo;
    }
    @Override
    public void despegar() {
        System.out.println("El avión despega.");
    }
    @Override
    public void volar() {
        System.out.println("El avión está volando.");
    }
    @Override
    public void aterrizar() {
        System.out.println("El avión aterriza.");
    }
    @Override
    public String toString() {
        return "Avión: Modelo=" + modelo;
    }
}
```

Ejercicio 1: Aeropuerto

```
// Clase Drone que implementa Volable
public class Drone implements Volable {
    private String nombre;
    public Drone(String nombre) {
        this.nombre = nombre;
    }
    @Override
    public void despegar() {
        System.out.println("El drone despega.");
    }
    @Override
    public void volar() {
        System.out.println("El drone está
volando.");
    }
    @Override
    public void aterrizar() {
        System.out.println("El drone aterriza.");
    }
    @Override
    public String toString() {
        return "Drone: Nombre=" + nombre;
    }
}
```

Ejercicio 1: Aeropuerto

Universidad Tecnológica del Perú

```
// Clase Aeropuerto
import java.util.ArrayList;
import java.util.List;
public class Aeropuerto {
    private List<Volable> vehiculosVolables;
    public Aeropuerto() {
        vehiculosVolables = new ArrayList<>();
    }
    public void agregarVehiculoVolable(Volable
vehiculoVolable) {
        vehiculosVolables.add(vehiculoVolable);
    }
    public void operarVehiculos() {
        for (Volable vehiculo : vehiculosVolables) {
            vehiculo.despegar();
            vehiculo.volar();
            vehiculo.aterrizar();
        }
    }
```

Ejercicio 1: Aeropuerto

```
public static void main(String[] args) {
        Aeropuerto aeropuerto = new Aeropuerto();
        Avion avion1 = new Avion("Boeing 747");
        Avion avion2 = new Avion("Airbus A380");
        Drone drone1 = new Drone("DJI Mavic Air");
        Drone drone2 = new Drone("Parrot Anafi");
        aeropuerto.agregarVehiculoVolable(avion1);
        aeropuerto.agregarVehiculoVolable(avion2);
        aeropuerto.agregarVehiculoVolable(drone1);
        aeropuerto.agregarVehiculoVolable(drone2);
        System.out.println("Operaciones en el
aeropuerto:");
        aeropuerto.operarVehiculos();
    }
}
```

Ejercicio 1: Aeropuerto

Practica

Implementa el ejercicio visto.

Conclusiones

- El procesamiento de colecciones de objetos en Java es esencial en la programación, ya que te permite gestionar datos de manera eficiente y realizar operaciones como búsqueda, ordenación y filtrado en una variedad de estructuras de datos.
- La elección de la estructura de datos adecuada y la comprensión de cómo procesarla son fundamentales para el desarrollo eficiente de aplicaciones en Java.

Cierre

¿Qué hemos aprendido hoy?

Universidad Tecnológica del Perú

Bibliografía

- MORENO PÉREZ, J. “Programación orientada a objetos”. RA-MA Editorial. https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=31933
- Vélez Serrano, José. “Diseñar y programar, todo es empezar: una introducción a la Programación Orientada a Objetos usando UML y Java”. Dykinson. https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=36368

Universidad Tecnológica del Perú