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

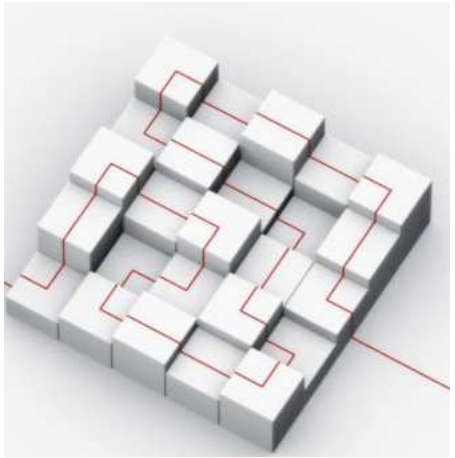
Universidad Tecnológica del Perú

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.



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.

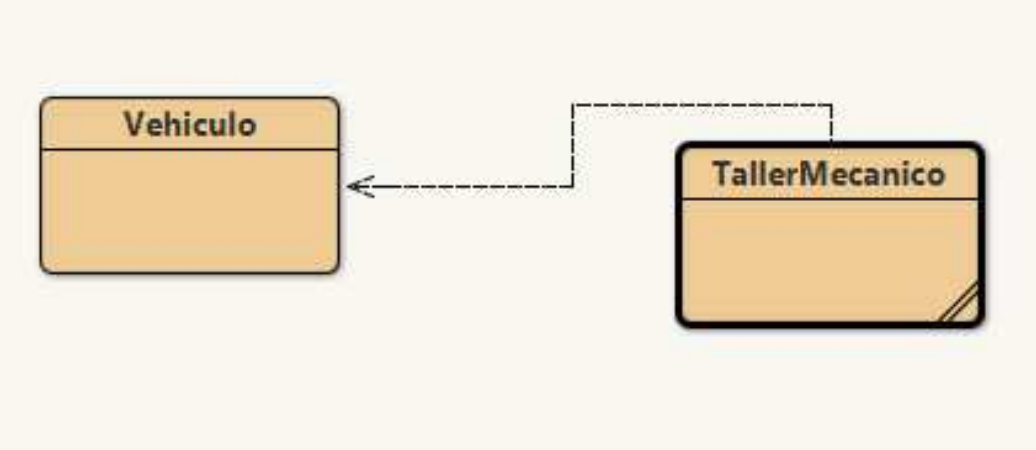
Iteradores

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

ArrayList

https://www.youtube.com/watch?v=MT77xTj76xE

Universidad Tecnológica del Perú



Ejercicio 1:
Taller Mecánica

```
public class Vehiculo {
    private String marca;
    private String modelo;
    public Vehiculo(String marca, String modelo) {
        this.marca = marca;
        this.modelo = modelo;
    }
    public String getMarca() {
        return marca;
    }
    public String getModelo() {
        return modelo;
    }
    @Override
    public String toString() {
        return "Vehículo: Marca=" + marca + ", Modelo=" + modelo;
    }
}
```

Ejercicio 1:
Taller Mecánica

Universidad Tecnológica del Perú

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
public class TallerMecanico {
    private List<Vehiculo> vehiculos;
    public TallerMecanico() {
        vehiculos = new ArrayList<>();
    }
    public void agregarVehiculo(Vehiculo vehiculo) {
        vehiculos.add(vehiculo);
    }
    public void mostrarVehiculos() {
        for (Vehiculo vehiculo : vehiculos) {
            System.out.println(vehiculo);
        }
    }
    public List<Vehiculo> buscarPorMarca(String marca) {
        List<Vehiculo> vehiculosEncontrados = new ArrayList<>();
        for (Vehiculo vehiculo : vehiculos) {
            if (vehiculo.getMarca().equalsIgnoreCase(marca)) {
                vehiculosEncontrados.add(vehiculo);
            }
        }
        return vehiculosEncontrados;
    }
```

Ejercicio 1: Taller Mecánica

Universidad Tecnológica del Perú

```
public static void main(String[] args) {
    TallerMecanico taller = new TallerMecanico();
    Scanner scanner = new Scanner(System.in);
    while (true) {
        System.out.println("\nMenu:");
        System.out.println("1. Agregar vehículo");
        System.out.println("2. Mostrar vehículos");
        System.out.println("3. Buscar vehículos por marca");
        System.out.println("4. Salir");
        System.out.print("Ingrese su opción: ");
        int opcion = scanner.nextInt();
        scanner.nextLine(); // Consumir la nueva línea
        switch (opcion) {
            case 1:
                System.out.print("Ingrese la marca del vehículo: ");
                String marca = scanner.nextLine();
                System.out.print("Ingrese el modelo del vehículo: ");
                String modelo = scanner.nextLine();
                Vehiculo vehiculo = new Vehiculo(marca, modelo);
                taller.agregarVehiculo(vehiculo);
                break;
```

Ejercicio 1: Taller Mecánica

Universidad Tecnológica del Perú

```
case 2:
                System.out.println("Lista de vehículos:");
                taller.mostrarVehiculos();
                break;
            case 3:
                System.out.print("Ingrese la marca a buscar: ");
                String marcaBusqueda = scanner.nextLine();
                List<Vehiculo> vehiculosEncontrados =
taller.buscarPorMarca(marcaBusqueda);
                System.out.println("Vehículos encontrados:");
                for (Vehiculo v : vehiculosEncontrados) {
                    System.out.println(v);
                }
                break;
            case 4:
                System.out.println("Saliendo del programa.");
                System.exit(0);
            default:
                System.out.println("Opción no válida.");
        }
    }
}
```

Ejercicio 1: Taller Mecánica

Universidad Tecnológica del Perú

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