UNIVERSIDAD TECNOLÓGICA DEL PERÚ

Facultad de Ingeniería

Programacion Orientada a Objetos

Sesión 10: Colecciones: LinkedList, ArrayList, Hashmap

Recordando:

¿Que vimos la clase pasada?

Universidad Tecnológica del Perú

Logro de aprendizaje

Al finalizar la sesión, el estudiante comprende varias estructuras de datos comunes, como listas, conjuntos, mapas y colas en la solución de problemas usando el lenguaje Java.

Universidad Tecnológica del Perú

Utilidad

- Permite al estudiante tomar decisiones informadas sobre qué tipo de colección usar en función de los requisitos de tu aplicación y las operaciones que necesitas realizar.
- Les permite aprender a agregar, eliminar, buscar, ordenar y manipular datos de manera efectiva utilizando las operaciones proporcionadas por las colecciones.

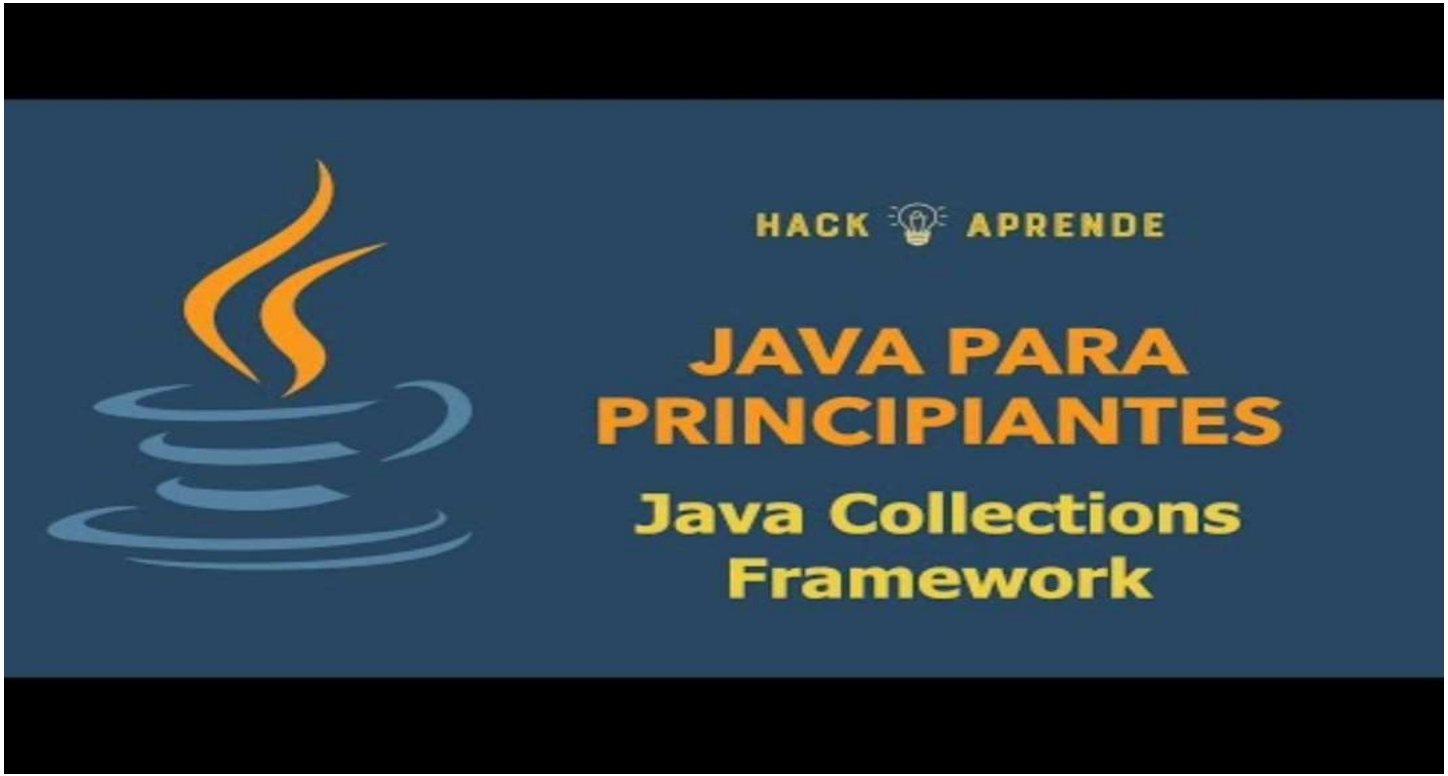
Agenda

- Concepto de Colecciones.
- Principales Colecciones:
  - LinkedList
  - Hashmap

Colecciones

- En Java, las colecciones son estructuras de datos que se utilizan para almacenar, manipular y gestionar conjuntos de objetos.
- Las colecciones son una parte fundamental de la programación en Java, ya que proporcionan formas eficientes y flexibles de organizar y trabajar con datos.
- Java ofrece una variedad de interfaces y clases para manejar colecciones, que se encuentran en el paquete java.util.
- A continuación, te explicaré las principales colecciones en Java y sus características:

Colecciones



https://www.youtube.com/watch?v=UUr_lhzk0vQ

Ejercicio de “LinkedList”

En este ejemplo, crearemos una lista de videojuegos disponibles en la tienda. Los clientes podrán agregar y eliminar juegos de su lista de deseos, y se les mostrará una lista de deseos actualizada. A continuación, tienes el código de muestra:

```
public class Videojuego {
    private String nombre;
    private double precio;

    public Videojuego(String nombre, double precio) {
        this.nombre = nombre;
        this.precio = precio;
    }

    public String getNombre() {
        return nombre;
    }

    public double getPrecio() {
        return precio;
    }

    @Override
    public String toString() {
        return nombre + " - $" + precio;
    }
}
```

Ejercicio de "LinkedList"

```
import java.util.LinkedList;
import java.util.Scanner;

public class TiendaVideojuegos {
    public static void main(String[] args) {
        LinkedList<Videojuego> listaDeDeseos = new LinkedList<>();
        listaDeDeseos.add(new Videojuego("The Legend of Zelda: Breath of the Wild", 49.99));
        listaDeDeseos.add(new Videojuego("Cyberpunk 2077", 39.99));
        listaDeDeseos.add(new Videojuego("Red Dead Redemption 2", 59.99));

        Scanner scanner = new Scanner(System.in);

        System.out.println("Bienvenido a la Tienda de Videojuegos");
        while (true) {
            System.out.println("\nLista de Deseos:");
            for (int i = 0; i < listaDeDeseos.size(); i++) {
                Videojuego juego = listaDeDeseos.get(i);
                System.out.println(i + 1 + ". " + juego.toString());
            }

            System.out.print("Elija una opción:\n" +
                    "1. Agregar juego a la lista de deseos\n" +
                    "2. Eliminar juego de la lista de deseos\n" +
                    "3. Salir\n" +
                    "Opción: ");
```

Ejercicio de "LinkedList"

```
int opcion = scanner.nextInt();

            switch (opcion) {
                case 1:
                    System.out.print("Ingrese el nombre del juego a agregar: ");
                    scanner.nextLine(); // Consumir el salto de línea anterior
                    String nombreJuego = scanner.nextLine();
                    System.out.print("Ingrese el precio del juego: ");
                    double precioJuego = scanner.nextDouble();
                    Videojuego nuevoJuego = new Videojuego(nombreJuego, precioJuego);
                    listaDeDeseos.add(nuevoJuego);
                    System.out.println("Juego agregado a la lista de deseos: " + nuevoJuego.toString());
                    break;
                case 2:
                    System.out.print("Elija el número del juego a eliminar: ");
                    int juegoAEliminar = scanner.nextInt();
                    if (juegoAEliminar >= 1 && juegoAEliminar <= listaDeDeseos.size()) {
                        Videojuego juegoEliminado = listaDeDeseos.remove(juegoAEliminar - 1);
                        System.out.println("Juego eliminado de la lista de deseos: " + juegoEliminado.toString());
                    } else {
                        System.out.println("Número de juego no válido.");
                    }
                    break;
                case 3:
                    System.out.println("Gracias por visitar la Tienda de Videojuegos.");
                    return;
                default:
                    System.out.println("Opción no válida. Por favor, elija una opción válida.");
            }
        }
    }
}
```

Universidad Tecnológica del Perú

Ejercicio de “HashMap”

En este ejemplo, crearemos una aplicación simple para gestionar el inventario de libros de una librería. Utilizaremos un HashMap para almacenar la información de los libros, donde la clave será el ISBN del libro y el valor será un objeto Libro que contendrá el título y el autor del libro. A continuación, tienes el código de muestra:

```
public class Libro {
    private String titulo;
    private String autor;

    public Libro(String titulo, String autor) {
        this.titulo = titulo;
        this.autor = autor;
    }

    public String getTitulo() {
        return titulo;
    }

    public String getAutor() {
        return autor;
    }

    @Override
    public String toString() {
        return "Título: " + titulo + ", Autor: " + autor;
    }
}
```

Ejercicio de “HashMap”

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class GestionLibreria {
    public static void main(String[] args) {
        Map<String, Libro> inventarioLibros = new HashMap<>();
        inventarioLibros.put("978-0134685991", new Libro("Clean Code", "Robert C. Martin"));
        inventarioLibros.put("978-0596009205", new Libro("Head First Java", "Kathy Sierra"));

        Scanner scanner = new Scanner(System.in);

        System.out.println("Bienvenido a la Librería");
        while (true) {
            System.out.println("\nMenú:");
            System.out.println("1. Mostrar inventario de libros");
            System.out.println("2. Agregar libro al inventario");
            System.out.println("3. Buscar libro por ISBN");
            System.out.println("4. Salir");
            System.out.print("Elija una opción: ");

 int opcion = scanner.nextInt();
            scanner.nextLine(); // Consumir el salto de línea

            switch (opcion) {
                case 1:
                    System.out.println("\nInventario de Libros:");
                    for (Map.Entry<String, Libro> entry : inventarioLibros.entrySet()) {
                        System.out.println("ISBN: " + entry.getKey() + ", " + entry.getValue());
                    }
                    break;
```

Universidad Tecnológica del Perú

Ejercicio de “HashMap”

```
case 2:
                System.out.print("Ingrese el ISBN del libro: ");
                String isbn = scanner.nextLine();
                System.out.print("Ingrese el título del libro: ");
                String titulo = scanner.nextLine();
                System.out.print("Ingrese el autor del libro: ");
                String autor = scanner.nextLine();
                Libro nuevoLibro = new Libro(titulo, autor);
                inventarioLibros.put(isbn, nuevoLibro);
                System.out.println("Libro agregado al inventario.");
                break;
            case 3:
                System.out.print("Ingrese el ISBN del libro a buscar: ");
                String isbnBusqueda = scanner.nextLine();
                Libro libroBuscado = inventarioLibros.get(isbnBusqueda);
                if (libroBuscado != null) {
                    System.out.println("Libro encontrado: " + libroBuscado);
                } else {
                    System.out.println("Libro no encontrado.");
                }
                break;
            case 4:
                System.out.println("Gracias por visitar la Librería.");
                return;
            default:
                System.out.println("Opción no válida. Por favor, elija una opción válida.");
        }
    }
}
```

DIFERENCIAS entre ARRAYLIST y LINKEDLIST en JAVA POO

https://www.youtube.com/watch?v=tLTbR9zkUfg

Practica

Implementa los ejercicios realizados.

Conclusiones

Las colecciones en Java son una parte esencial del desarrollo de software y pueden proporcionar numerosos beneficios en términos de organización de datos, eficiencia y legibilidad del código. Sin embargo, es importante utilizarlas de manera adecuada y consciente para aprovechar al máximo sus ventajas.

Cierre

¿Qué hemos aprendido hoy?

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