

Diplomado en Programación Java

UAPA - MESCYT

Lab04: Tipos de datos enumerados y Colecciones

Por Eudris Cabrera Rodríguez

Objetivo:

Practicar los conceptos de tipos de datos enumerados y las colecciones en Java.

Fecha de Realización: 01 Diciembre del 2018

*Nota: Tomar como base el proyecto **javase_enum_colecciones**.*

1- Se tiene un enum llamado Fruta que contiene un listado de frutas.

```
public enum Fruta {  
    MANGO, NARANJA, MELON, ZAPOTE, FRESA, SANDIA  
}
```

En la clase PruebaEnum se desea saber si te gusta una fruta en específico.

```
public class PruebaEnum {  
    Fruta fruta;  
  
    public PruebaEnum(Fruta fruta) {  
        this.fruta = fruta;  
    }  
  
    public void decirSiTeGusta(){  
  
    }  
  
    public static void main(String[] args) {  
  
        PruebaEnum frutaMango = new PruebaEnum(Fruta.MANGO);  
        frutaMango.decirSiTeGusta();  
  
        PruebaEnum frutaNaranja = new PruebaEnum(Fruta.NARANJA);  
        frutaNaranja.decirSiTeGusta();  
  
    }  
}
```

}

En el método **decirSiTeGusta()**, agregar lógica para que cumpla con la siguiente salida.

Salida:

Mango es muy bueno.

No me gusta la sandía.

La naranja es muy buena pero prefiero el zapote.

El melón es mi favorito.

Agregar en el método **main(String[] args)** las demás opciones que completan la salida.

2- Se desea crear un enum llamado "Entrada" que contemple los precios de las entradas a un evento en un sala famosa de entretenimiento.

Los precios son los siguientes 1,000 Balcón, 1,500 Platea, \$2,500 VIP y RD\$3,000 Front Stage Platinum.

Crear una nueva clase llamada "**PruebaEntrada**" y recorriendo todos los valores del enum entrada permite saber el listado de precios.

Tomar en consideración el método "**values**".

Salida:

El precio de la entrada en "**BALCON**" es RD\$1,000.

El precio de la entrada en "**PIATEA**" es RD\$1,500.

El precio de la entrada en "**VIP**" es RD\$2,500.

El precio de la entrada en "**FRONT_STAGE_PLATINUM**" es RD\$3,000.

Tomar en consideración las reglas para nombrar las constantes.

3- Dado un listado de personas crear métodos en la clase **CriterioBusqueda** que permita lista todos los elementos de la lista, filtrar entre las edades 25 y 50 años, y los mayores de 50 personas.

Nota: *Tomar como base el proyecto **javase_colecciones_map_set***

4- **CalculadoraPresupuesto** es un programa que recibe del usuario el monto presupuestado a gastar en un mes, luego toma todos los gastos fijos que tiene el usuario y saca un balance de lo presupuestado versus lo gastado. Si el balance es menor que cero, el sistema despliega el siguiente mensaje. "**Ha excedido lo presupuestado por RD\$xxx.x pesos**". Si el balance es mayor de cero el sistema despliega el siguiente mensaje "**Todavía está dentro de lo presupuestado, puedes gastar RD\$xxx.x pesos**". Si el balance es igual a cero, el sistema despliega el siguiente mensaje "**Gastaste exactamente lo presupuestado**".

Usando la clase **CalculadoraPresupuesto** ubicada en mencionado proyecto, escribir programa

que cumpla con el escenario planteado. El programa termina al escribir un valor negativo.

Sustituir xxx.x por valores reales.

5- Dado la siguiente lista

```
List<String> lista = Arrays.asList("Edificio", "Casa", "Escuela", "Apartamento", "Carro",  
"Tienda", "Farmacia", "Supermercado", "Plaza", "Cine", "Hotel" );
```

Escribir programa que tome de la lista todas las palabras que tengan longitud par y se agreguen a un set.

```
Set<String> pares = new HashSet<String>();
```

Luego recorrer el set y desplegar el contenido por la salida estándar.

6- Dado la siguiente lista

```
List<Persona> personas = new ArrayList<>();
```

```
personas.add(new Persona("Rita", "Perez", Genero.FEMENINO, 15));  
personas.add(new Persona("Maria", "Rosa", Genero.FEMENINO, 25));  
personas.add(new Persona("Luz", "Santos", Genero.FEMENINO, 30));  
personas.add(new Persona("Juan", "Perez", Genero.MASCULINO, 55));  
personas.add(new Persona("Victor", "Martinez", Genero.MASCULINO, 32));  
personas.add(new Persona("Rafael", "Santos", Genero.MASCULINO, 47));  
personas.add(new Persona("Manuel", "Cabral", Genero.MASCULINO, 0));  
personas.add(new Persona("Marta", "Rodriguez", Genero.FEMENINO, 18));  
personas.add(new Persona("Andres", "Jimenez", Genero.MASCULINO, 52));  
personas.add(new Persona("Hector", "Perez", Genero.MASCULINO, 51));
```

Convertir los elementos de la lista a un Map, donde la llave (key) es la concatenación de la posición del elemento dentro de la lista y el nombre y apellido.

```
Map<String, Persona> mapPersonas = new HashMap<String,Persona>();
```

7-Dado la siguiente lista

```
List<String> lista = Arrays.asList("Edificio", "Casa", "Escuela", "Apartamento", "Edificio",  
"Tienda", "Farmacia", "Apartamento", "Plaza", "Farmacia", "Hotel" );
```

Escribir programa que retorne un HashMap que contenga la palabra y la cantidad de veces que aparece en la lista.

8- Dado la siguiente lista

```
List<String> lista = Arrays.asList("Edificio", "Casa", "Escuela", "Apartamento", "Carro",  
"Tienda", "Farmacia", "Supermercado", "Plaza", "Cine", "Hotel" );
```

Escribir programa que retorne la palabra de mayor longitud dentro de la lista;

9- Combustible App es una herramienta que nos permite visualizar los precios de los combustibles.

Dado el proyecto **java_combustible_app** se desea realizar lo siguiente:

Que el usuario digite algunas de las siguientes palabras, gasolina, gasoil, glp, gnv, kerosene y esta le muestre el precio de dicho combustible.

Actualmente muestra un listado de todos los combustibles, se desea que solo muestre el deseado.

Ejemplo:

Datos de Entrada y salida:

Bienvenido a Combustible App

Escriba el tipo combustible: glp

Descripción: Gas Licuado de Petróleo (GLP)

Precio: 83.20

Continuar ? (y/n): y

Si el usuario digita una opción inválido debe mostrar mensaje diciendo que es una opción inválida.

Además si el usuario digita el tipo de combustible en mayúscula o minúscula debe presentar en pantalla la opción seleccionada.

Consejo:

Ya se tiene la forma de capturar el parámetro digitado, usted debe crear la lógica en nuestra fuente de datos **CombustibleDB** que pasado como parámetro el texto digitado, permita filtrar por el mismo.