Material descargable

Programación **orientada a objetos**

egg

Introducción

En la vida real, a menudo encontramos situaciones en las que necesitamos trabajar con conjuntos de datos que son constantes, es decir, valores que no cambian y son conocidos de antemano. Estos datos constantes pueden ser opciones, categorías, estados, tipos o cualquier otro conjunto predefinido de valores.

Por ejemplo, considera una aplicación de gestión de pedidos en línea. Tendrás diferentes estados para un pedido, como "pendiente", "en proceso", "enviado" o "entregado". Estos estados son constantes y se utilizan en todo el sistema para representar el estado actual de cada pedido.

En estos casos, es importante tener una forma de representar y manejar estos conjuntos constantes de datos de manera eficiente y segura. **Aquí es donde entran en juego los enums en Java.**

Enums

En Java, un enum (enumeración) **es un tipo de datos especial que permite definir una colección de constantes nombradas.** Los enums son una forma más segura y legible de representar un conjunto fijo de valores en comparación con el uso de constantes enteras o cadenas de texto.

Para declarar un enum en Java, se utiliza la palabra clave *enum*, *seguida del nombre del enum y los valores constantes entre llaves*.

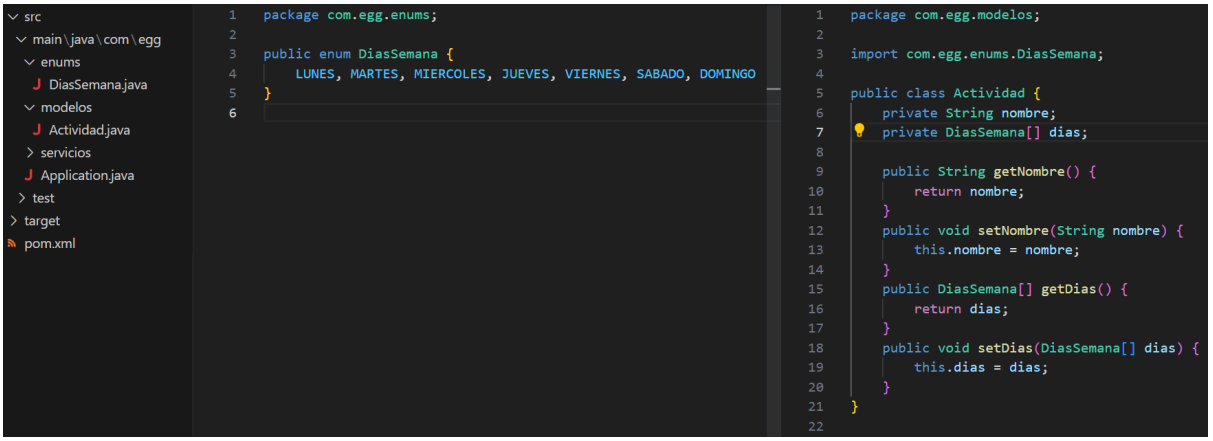
Aquí tienes un ejemplo básico de cómo se declara un enum llamado **DiaSemana** que representa los días de la semana:

```java
public enum DiaSemana {
    LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO
}
```

Puedes declarar un Enum dentro de una clase de la siguiente manera:

```java
public class Actividad {
    private String nombre;
    private DiaSemana[] dias;
    public enum DiaSemana {
            LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO
    }
}
```

Dado que el ejemplo anterior no es considerada de las mejores prácticas, sería mejor declararlo en un paquete “enums” como un archivo separado:



En este caso, DiaSemana es el nombre del enum y las constantes dentro del enum son los días de la semana. Los nombres de las constantes son en mayúsculas por convención.

Enums con propiedades

Los enums en Java son tratados como clases especiales. Por lo tanto, pueden tener métodos, campos y constructores. Aquí tienes un ejemplo de cómo agregar métodos y un campo al enum DiaSemana:

```java
public enum DiaSemana {
    LUNES("Primer día de la semana"),
    MARTES("Segundo día de la semana"),
    MIERCOLES("Tercer día de la semana"),
    JUEVES("Cuarto día de la semana"),
    VIERNES("Quinto día de la semana"),
    SABADO("Sexto día de la semana"),
    DOMINGO("Séptimo día de la semana");

    private String descripcion;

    private DiaSemana(String descripcion) {
        this.descripcion = descripcion;
    }

    public String getDescripcion() {
        return descripcion;
    }
}
```

En este ejemplo, se agrega un campo descripción al enum DiaSemana y un constructor que permite asignar un valor a este campo para cada constante del enum. También se define un método getDescripcion() para obtener la descripción de cada día de la semana, pero no se crea el método setDescription(), no solo porque al ser una constante no debería cambiar sus propiedades, sino también porque no se puede.

Las constantes de un enum y sus propiedades se cargan de manera estática En Java, los enums se consideran tipos de datos finales y se definen como conjuntos fijos de valores constantes en tiempo de compilación. Esto significa que una vez que se ha definido un enum y se ha asignado un conjunto de constantes, no se pueden agregar, eliminar o modificar esas constantes durante la ejecución del programa.

Manipulación de Enums

Aunque no se puedan modificar las constantes de un Enum, puedes realizar varias operaciones y manipulaciones con los enums en Java. Aquí hay algunas formas de trabajar con ellos:

- **Acceder a las constantes del enum:** Puedes acceder a las constantes de un enum utilizando el nombre del enum seguido del nombre de la constante. Por ejemplo, si tienes un enum llamado "DiaSemana" con las constantes "LUNES", "MARTES", etc., puedes acceder a ellas de la siguiente manera.

```
public class Application {
    public static void main(String[] args) {
        DiaSemana dia = DiaSemana.LUNES;
    }
}
```

- **Obtener todos los valores del enum:** Puedes obtener un arreglo de todas las constantes de un enum utilizando el método values(). Por ejemplo: Esto te dará un arreglo que contiene todas las constantes del enum.

```
public class Application {
    public static void main(String[] args) {
        DiaSemana[] dias = DiaSemana.values();
    }
}
```

- **Comparar valores de enum:** Puedes comparar valores de enum utilizando el operador de igualdad (==).

```
public class Application {
    public static void main(String[] args) {
        DiaSemana dia = DiaSemana.LUNES;
        if (dia == DiaSemana.LUNES) {
            System.out.println(dia);
        }
    }
}
```

- **Utilizar enums en estructuras de control:** Puedes utilizar un enum en una estructura de control, como un switch, para tomar decisiones basadas en el valor del enum.

```
public class Application {
    public static void main(String[] args) {
        DiaSemana dia = DiaSemana.LUNES;
        String tipoDia = switch (dia) {
            case LUNES, MARTES, MIERCOLES, JUEVES, VIERNES -> "Día
laboral";
            case SABADO, DOMINGO -> "Fin de semana";
        };
        System.out.println(tipoDia);
    }
}
```