Los enum son implícitamente final, lo que significa que no se pueden extender. Esto evita que se creen subclases de un enum, asegurando que los enum son tipos completos y no pueden modificarse a través de la herencia. Por la misma razón, un enum no Enums son Implícitamente final: puede ser sealed ya que esto implicaría la posibilidad de tener subclases permitidas. Los enum implementan java.lang.Comparable, lo que permite que los enum se puedan ordenar en colecciones como SortedSet, TreeSet, y TreeMap. Implementación de El orden natural de los enum se Comparable: basa en el orden de declaración de sus constantes, que corresponde a su valor ordinal. ordinal() devuelve el índice (empezando desde 0) de la constante en la declaración del Características Adicionales de Conceptos Clave de enum. Este índice representa la java.lang.Enum Enums en Java posición en la que se definió la Método ordinal(): constante. name() devuelve el nombre exacto de la constante del enum tal como se declara en el código. Este nombre es Método name(): inmutable y no cambia. values(): Devuelve un array con todas las constantes del enum. Este método es útil para iterar sobre todas las constantes. valueOf(String): Busca una Los enum tienen dos métodos constante del enum que estáticos proporcionados coincida exactamente con el automáticamente por el Métodos Generados nombre proporcionado como compilador: Automáticamente: argumento (distingue entre mayúsculas y minúsculas). Si static no encuentra coincidencia,

El constructor de un enum es siempre private, y no puede ser public, default y protected. Esto asegura que las instancias de los enum solo se puedan crear dentro de la propia declaración del enum.

Constructor Privado:

Si un enum no declara explícitamente un constructor, el compilador proporciona automáticamente un constructor private sin parámetros.

Por defecto, el método toString() de un enum imprime el nombre de la constante tal como se declara en el enum. Sin embargo, se puede sobrescribir para personalizar la salida.

Método toString():

Un enum no puede extender otra clase o enum porque todos los enum extienden implícitamente java.lang.Enum. Esto establece las reglas y comportamientos básicos de los enum en Java.

Restricciones de Herencia:

Sin embargo, los enum pueden implementar interfaces, lo que permite añadir comportamiento adicional a los tipos de enum.

Los enum mantienen exactamente una instancia de cada una de sus constantes, lo cual asegura que son únicas y no pueden duplicarse.

Imposibilidad de Clonar:

No se puede sobrescribir el método clone() en un enum ya que java.lang.Enum lo define como final.

Se pueden definir métodos abstract

abstract

Cada enum debe sobreescribir el método abstract

Se pueden definir atributos y métodos static

Se puede definir el método main

DIASEMANA ds = DIASEMANA.valueOf("LUNE");

lanza una

IllegalArgumentException.