Eduardo P Cabrera Rendon
3 “A”

Equals

El método equals es un método de la clase Object en Java que se utiliza para comparar dos objetos para determinar si son iguales o no.
El propósito principal de este método es comparar el contenido de dos objetos para determinar si son equivalentes, en lugar de simplemente comparar las referencias de los objetos. La implementación predeterminada del método equals en la clase Object compara las referencias de los objetos, lo que significa que dos objetos se consideran iguales solo si son la misma instancia en la memoria.
Sin embargo, en la mayoría de las clases que creas en Java, es recomendable anular (override) el método equals para proporcionar una implementación personalizada que compare el contenido de los objetos según tus necesidades. Al hacerlo, puedes definir qué significa que dos objetos de tu clase sean iguales.

hashCode

El método hashCode de la clase Object en Java se utiliza para obtener un valor hash numérico de un objeto. Este valor hash es un número entero que se usa principalmente en estructuras de datos basadas en hash, como HashMap, HashSet, y Hashtable, para acelerar la búsqueda y recuperación de objetos.
El valor hash se calcula a partir de los datos internos del objeto y debe ser coherente con el método equals. Es decir, si dos objetos son iguales según el método equals, entonces sus valores hash deben ser iguales. Esto garantiza que los objetos que son iguales se almacenen en la misma "casilla" (bucket) de una estructura de datos basada en hash, lo que mejora el rendimiento de las operaciones de búsqueda y recuperación.

toString

El método toString es un método de la clase Object en Java que se utiliza para obtener una representación en forma de cadena (string) de un objeto.
Este método se utiliza para proporcionar una representación legible por humanos de un objeto, lo que es especialmente útil para la depuración y para la impresión de información sobre un objeto en registros o salidas de la aplicación. Por defecto, la implementación del método toString en la clase Object devuelve una cadena que contiene el nombre de la clase del objeto, seguido de un signo "@" y el valor hexadecimal de la dirección de memoria del objeto. Esta representación no es muy útil en la mayoría de los casos.
Sin embargo, es una buena práctica anular (override) el método toString en tus propias clases personalizadas para proporcionar una representación más significativa y útil de tus objetos. Puedes diseñar la representación de la cadena de acuerdo con tus necesidades, incluyendo información sobre los atributos y el estado del objeto, lo que facilita la depuración y la comprensión de la información que contienen los objetos.}

Clone

El método clone de la clase Object en Java se utiliza para crear una copia superficial de un objeto. Una copia superficial significa que se crea un nuevo objeto, pero los campos de datos del nuevo objeto contienen referencias a los mismos objetos que los campos de datos del objeto original, en lugar de crear copias independientes de esos objetos. Esto puede ser útil en algunos casos, pero debes tener cuidado al utilizarlo, ya que puede llevar a resultados inesperados si los objetos contenidos en el objeto original son mutables.
Ten en cuenta que el método clone en la clase Object es protegido, lo que significa que solo puede ser accedido por subclases de Object o por clases que estén en el mismo paquete. Además, el objeto que se clona debe implementar la interfaz Cloneable. Si no implementa esta interfaz y se intenta clonar el objeto, se lanzará una excepción CloneNotSupportedException