Síntesis conceptual

Grado: Administración de sistemas informáticos en red
Asignatura: Programación básica
Unidad: 8. Estructura de almacenamiento Arrays y cadenas de caracteres.

Resumen

Cuando se trabaja con una agrupación de datos del mismo tipo es necesario tener una herramienta para poder almacenarlo en memoria y usarlos en nuestro algoritmo. Las arrays permiten agrupar y almacenar más de un elemento en memoria del mismo tipo primitivo o abstracto con un límite fijo de valores. Los arrays son objetos estáticos y secuenciales, no es posible cambiar su tipo de datos y su tamaño para incrementarlo o decrementarlo en ejecución. La representación de un array puede ser por dimensiones, unidimensional representa una sola dimensión como si fuera una fila de elementos, en cambio multidimensional, con más de una dimensión representa una matriz o tabla.

Para el uso de las arrays es necesario trabajar con un índice o referencia que permite conocer la posición del array. El índice que puede tener un array es de 0 a N-1, N representa la longitud máxima que puede tener. Con el uso del índice se aprenderá diferentes funcionalidades sobre un array, como hacer un recorrido completo a todos sus elementos, cambiar algún elemento, añadir, buscar, etc. Para facilitar el trabajo con los arrays se hará uso de funcionalidades de la clase *java.util.Arrays* perteneciente a la librería de Java.

Una cadena de caracteres es una agrupación de datos almacenados en un array de tipo char. Para facilitar el tratamiento de la cadena de caracteres, en java se desarrolló el tipo String para no tener que hacer uso de arrays de caracteres. Aún que puede causar confusión String no es un tipo primitivo, se puede observar que se declara con su primer carácter en mayúscula. Haciendo uso del tipo String nos permite realizar diferentes operaciones sobre la cadena de caracteres, como, por ejemplo, conocer la longitud de la cadena, búsqueda y comparación de algún carácter, palabra o frase; Remplazar algún valor o transformarlo a mayúsculas o minúsculas, etc.

Otras de las funcionalidades de java es poder transformar de un tipo de datos primitivo a otro, o incluso a tipo String. Para poder realizar la conversión conoceremos la funcionalidad *.parse* de las clases envoltorio de los tipos primitivos o la funcionalidad *.valueOf* para tipos String.

Conceptos fundamentales

- **Array**: Objeto estático que almacena o contiene elementos de forma secuencial con un número fijo de valores del mismo tipo, primitivos o abstractos.

- **Dimensional:** Indica la profundidad que puede tener un array. Puede ser unidimensional o multidimensional.

- **null**: Palabra reservada para indicar vacío o la nada.

- **new:** Palabra reservada para instanciar un nuevo objeto.
- **java.util.Arrays:** Clase que contiene funcionalidad para operar con arrays.
- **copia *deep* o profunda**: Método de copiado para hacer una copia exacta de los mismos valores. El elemento copiado como el original serán independientes.
- **copia *shallow* o poco profunda:** Método de copiado para hacer una copia exacta de las direcciones de memoria que ocupan el original, no de los valores. El elemento copiado como el original apuntarán a la misma dirección, cualquier cambio afectará a las dos partes.
- **String:** Tipo de datos no primitivo para representar una cadena de caracteres.
- **Concatenar:** Unir el valor de varios elementos.