



APRENDERAPROGRAMAR.COM

COLECCIONES DE OBJETOS
DE TAMAÑO VARIABLE EN
JAVA. COLLECTION, LIST,
SET, QUEUE, MAP, ETC.
(CU00664B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº64 curso Aprender programación Java desde cero.

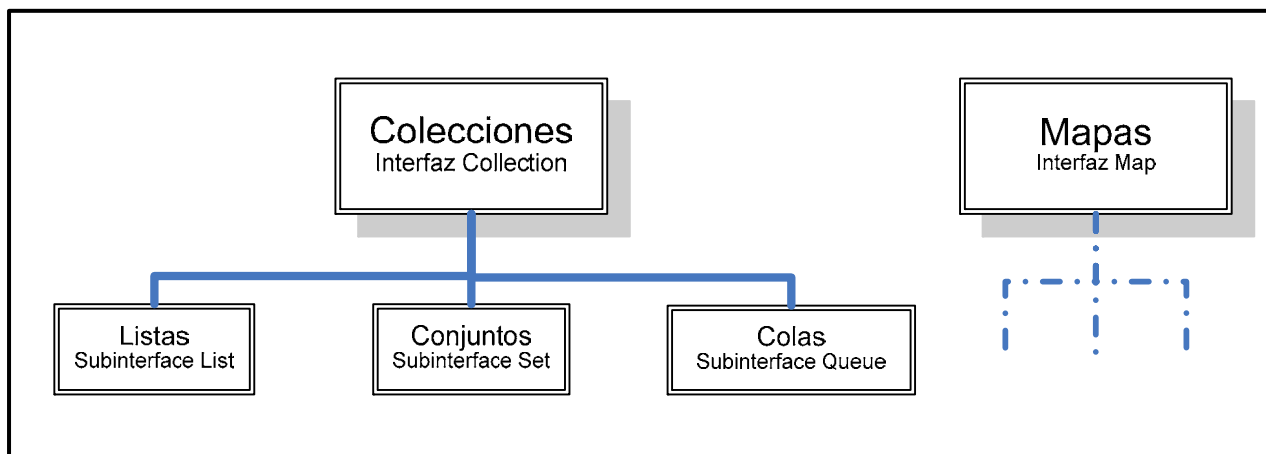
Autor: Alex Rodríguez

COLECCIONES DE OBJETOS DE TAMAÑO VARIABLE. CONTENEDORES.

Una colección de objetos es un objeto que puede almacenar un número variable de elementos siendo cada elemento otro objeto. Siguiendo con nuestro símil, podríamos ver una colección como una caja que contiene más cajas dentro. Puede haber distintos tipos de colecciones de tamaño “flexible”, es decir, que se pueden encoger o agrandar según las necesidades.



En Java se distinguen diversos tipos, agrupados en una estructura general similar a la que muestra este esquema.



Por el momento consideraremos que una interfaz viene siendo algo así como una protoclasa: define ciertas cosas que van a compartir cierto número de subinterfaces y de clases.

Cada tipo de colección tiene unas características.

- a) **Una lista (list)** es una colección de objetos donde cada uno de ellos lleva un índice asociado. Así, podríamos tener una lista con los nombres de las personas que han utilizado un servicio de acceso a internet que podría ser: usuarios --> (Juan R.R., Sara G.B., Rodolfo M.N., Pedro S.T., Claudio R.S., Juan R.R.). Donde cada contenido va asociado a un índice, usuario(0) sería Juan R.R., usuario(1) sería Sara G.B, usuario(2) sería Rodolfo M.N. y así sucesivamente. En una lista podemos insertar y eliminar objetos de posiciones intermedias. Ejemplos de listas son la clase *ArrayList* y *LinkedList* del API de Java.

- b) **Un conjunto (set)** sería una colección de objetos que no admite duplicados. Siguiendo el ejemplo anterior, un conjunto nos serviría para saber los usuarios distintos que han utilizado el servicio de acceso a internet, pero no tendríamos información sobre el orden y una misma persona no aparecería más de una vez, ni siquiera aunque hubiera utilizado el servicio varias veces. Ejemplo de conjunto sería la clase *HashSet* del API de Java.
- c) **Una cola (queue)** sería una colección de objetos que se comportan como lo haría un grupo de personas en la cola de una caja de un supermercado. Los objetos se van poniendo en cola y el primero en salir es el primero que llegó.
- d) **Una pila (stack)** sería una colección de objetos que se comportan como lo haría un montón de objetos apilados, el último en añadirse queda en la cima de la pila y el primero en salir es el último que ha llegado. El primero en llegar está en el fondo de la pila.
- e) **Existen otros tipos de colecciones.**

Lo cierto es que las estructuras de datos en Java ofrecen muchas posibilidades y variantes. Por ejemplo, podemos tener conjuntos sin orden entre los elementos, pero también conjuntos ordenados. Y podemos tener colas donde el objeto que sale primero no es el primero que llegó, sino el que tiene mayor prioridad o “urgencia” por salir. Para escoger un tipo de comportamiento u otro existen distintas clases que se catalogan como colecciones en el API de Java. Incluso existen clases que sirven para agrupar objetos que Java no clasifica como colecciones, aunque su funcionamiento es muy similar. Sería el caso de los Maps, objetos que contienen parejas de objetos donde un elemento sirve para encontrar al otro, al igual que un listín telefónico contiene parejas “Nombre de personas – Datos de dirección y teléfono” y el nombre de persona es lo que nos sirve para encontrar sus datos asociados. En la nomenclatura de Java, los Maps no son colecciones. Por eso a veces se usa el término “contenedores de objetos” para hacer referencia a listas, conjuntos, colas, mapas, etc. que al fin y al cabo son objetos que contienen más objetos (como una caja que contiene más cajas).

¿Cómo saber qué clase elegir? Hay varios factores a tener en cuenta, entre ellos el número de datos que tenemos que gestionar (no es lo mismo trabajar con una colección de 50 objetos que con una colección de 50.000 objetos) y el tipo de procesos que tenemos que realizar con ellos (no es lo mismo una lista en que los nuevos elementos se añaden casi siempre al final de la lista que una lista donde los nuevos elementos se añaden frecuentemente en posiciones intermedias). Cada clase resulta más eficiente que otra para realizar determinados procesos. Esto es de especial interés cuando hay que gestionar muchos datos. Si hablamos de sólo unas decenas de datos no vamos a ser capaces de apreciar diferencias de rendimientos.

Nosotros no vamos a estudiar todas las clases contenedoras de objetos porque excedería los objetivos de este curso. Citaremos simplemente algunas clases que son frecuentemente usadas por los programadores porque sirven para cubrir muchas necesidades de programación. Estas clases son *ArrayList*, *LinkedList*, *HashSet* y *HashMap*. Vamos a centrarnos primeramente en una de las clases más usadas, la clase *ArrayList* del API de Java. Estudiando esta clase aprenderemos cosas que después nos serán útiles para aplicar a cualquier contenedor de objetos. No obstante, siempre habremos de consultar la documentación del API de Java para conocer a fondo una clase.

EJERCICIO

Consulta la clase Stack en el API de Java. ¿Cómo se llama el método para consultar el objeto en la cima de la pila? ¿Cómo se llama el método para consultar si la pila está vacía? ¿El método pop podríamos clasificarlo como tipo procedimiento, como tipo función, o como tipo mixto (procedimiento y función)?

Próxima entrega: CU00665B

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188