

INTRODUCCIÓN A LAS COLECCIONES Y TIPOS GENÉRICOS EN JAVA

Introducción

Este pequeño artículo tiene por objeto introducirnos en el uso de las colecciones de objetos y de los tipos genéricos. En particular se explicará como usar los *ArrayList* para almacenar cualquier tipo de objeto. Podéis consultar la sección Referencias para ampliar la información.

Las colecciones son ni más ni menos que contenedores dinámicos de objetos. Su tamaño varía en tiempo de ejecución en función de las inserciones y extracciones de objetos a la misma. Las diferencias entre las distintas colecciones radican principalmente en dos aspectos:

1. Su **interfaz**, que podemos entender, de forma más o menos simplificada, como los métodos públicos accesibles por el usuario. Por ejemplo, el API *Collection* proporciona el interfaz *List* para implementar listas o secuencias y *Set* para implementar conjuntos de objetos. La diferencia entre ambos es que un *Set* no puede contener objetos duplicados mientras que un *List* sí.
2. Su **implementación**. Las implementaciones de los métodos del interfaz *List* para las colecciones *ArrayList* y *LinkedList* son diferentes. Entre otras diferencias, *ArrayList* es lo más parecido a un array, ya que permite el acceso a cualquier posición de la colección en un tiempo constante. Si por el contrario, estamos insertando y extrayendo objetos constantemente en la lista, es conveniente el uso de un *LinkedList*.

Todas las clases relacionadas con el API *Collection* de Java se incluyen en el paquete **java.util**.

El soporte de genéricos (conocidos como plantillas o *templates* en C++) apareció en la versión 5 de Java para solucionar el problema de las colecciones que gestionaban los objetos a través del tipo *Object*. Los problemas que ocasionaban este tipo de gestión es que había que realizar conversiones explícitas en la recuperación de objetos por lo que el compilador no realizaba ninguna comprobación. Con el uso de genéricos esto no ocurre, evitando la aparición de problemas durante la ejecución.

Uso de ArrayList con genéricos

Para declarar una colección de objetos genéricos se utiliza la siguiente expresión:

```
tipo_coleccion <tipo_objeto> nombreVariable;
```

Por ejemplo, declararemos la variable *listaAlumnos* de tipo *ArrayList* de cadenas de caracteres sería:

```
ArrayList <String> listaAlumnos;
```

Para crear un objeto de la colección se haría según la expresión:

```
nombreVariable = new tipo_coleccion<tipo_objeto>();
```

Para el ejemplo de la lista de alumnos:

```
listaAlumnos = new ArrayList<String>();
```

Si en nuestra lista de alumnos tratamos de insertar un objeto de distinto tipo al de la colección el compilador generaría un error. También, es conveniente aclarar que es posible el polimorfismo en las colecciones por lo que podemos incluir en la colección, objetos que son subclases del tipo definido para la colección.

Resumiremos los métodos principales para *ArrayList*. El parámetro *E* indica el tipo genérico para el que está definida la colección y que es sustituido por el compilador.

- *boolean add(E e)*. Inserta el objeto al final de la lista y devuelve el valor true.
- *boolean add(int indice, E e)*. Añade el objeto en la posición indicada desplazando el resto de objetos, si los hubiera, una posición hacia la derecha.
- *E get(int indice)*. Obtiene el objeto que se encuentra en la posición indicada en el parámetro pero no se borra de la lista.
- *E remove(int indice)*. Elimina el objeto que se encuentra en la posición. El método devuelve el objeto eliminado.
- *int size()*. Devuelve el número de elementos en la lista.
- *void clear()*. Elimina todos los elementos de la lista.

Referencias

La lista completa de métodos está disponible en la documentación del API de Java:

<http://java.sun.com/javase/6/docs/api/>

Para una información más completa del API Collection de Java puedes consultar el siguiente enlace:

http://www.javahispano.org/contenidos/es/introduccion_al_api_collections/

Como manual de referencia existe este tutorial de SUN sobre Collections (ENG):

<http://java.sun.com/docs/books/tutorial/collections/index.html>