



APRENDERAPROGRAMAR.COM

EL PAQUETE JAVA.UTIL DEL
API JAVA. PRINCIPALES
INTERFACES Y CLASES:
STRINGTOKENIZER, DATE,
CALENDAR, HASHSET,
TREEMAP, TREESET...
(CU00916C)

Sección: Cursos

Categoría: Lenguaje de programación Java nivel avanzado I

Fecha revisión: 2039

Resumen: Entrega nº16 curso "Lenguaje de programación Java Nivel Avanzado I".

Autor: Manuel Sierra y José Luis Cuenca

EL PAQUETE JAVA.UTIL

El paquete java.util, es uno de los principales paquetes que usaremos cuando estemos programando con el lenguaje Java, y junto al paquete java.lang es uno de los más utilizados en cualquier tipo de desarrollo informático basado en Java. Este paquete, sobre el que hablaremos a continuación, es un conjunto bastante grande tanto de interfaces como de clases de diversos tipos, que en conjunto podrían llamarse "utilidades".



DESCRIPCIÓN DEL PAQUETE JAVA.UTIL

Dado que el paquete java.util reúne un gran número de clases e interfaces, vamos a describir a continuación las que consideramos más importantes y usadas en base a nuestra experiencia. Evidentemente podremos aprovechar la documentación oficial (instalada en nuestro equipo o disponible vía internet) para poder consultar por nuestra cuenta el resto de interfaces y clases del paquete que deseemos.

A continuación vamos a describir en primer lugar las interfaces de este paquete con las que vamos a tratar durante este curso principalmente.

Las principales interfaces serán:

Interfaces en java.util
Collection
Comparator
Iterator
List
Map
Set
SortedMap
SortedSet

Mientras que las principales clases serán:

Clases en java.util
Calendar
Date (*)
HashMap
HashSet
LinkedList
StringTokenizer
TreeMap
TreeSet

(*) Date: su uso ha dejado de estar recomendado ya que las últimas versiones de Java incorporan nuevas clases para el tratamiento de fechas.

BREVE DESCRIPCIÓN DE LAS INTERFACES DE JAVA.UTIL

A continuación vamos a comentar brevemente cada una de las interfaces descritas. Estas las veremos un poco más adelante con ejemplos y ejercicios resueltos.

COLLECTION

Esta interfaz es la raíz de todas las interfaces relacionadas con colecciones de elementos. Algunas colecciones pueden admitir duplicados de elementos dentro de ellas, otras no. Algunas colecciones pueden tener los elementos ordenados, otras no. El JDK no proporciona ninguna implementación de esta interfaz y son respectivamente sus subinterfaces las que implementarán sus métodos como son por ejemplo la interfaz Set o List.

COMPARATOR

Similar a la interfaz Comparable que ya conocemos del paquete java.lang (que definía el método compareTo) nos define métodos para poder comparar 2 elementos y poder ordenarlos (los métodos compare y equals).

ITERATOR

Nos define los métodos que tenemos que implementar para hacer un iterador sobre una colección de elementos. Esto en parte fue visto anteriormente cuando vimos la implementación de la interfaz del paquete java.lang Iterable. Este nos obligaba a tener un método llamado iterator(), que nos devolvía un objeto de la interfaz Iterator.

LIST

Nos define el comportamiento de una lista de elementos. Esta lista normalmente admite elementos duplicados o iguales y son principalmente accedidos por su posición o por su secuencia.

MAP, SET, SORTEDMAP Y SORTEDSET

La interfaz Map, nos definirá el comportamiento de un objeto que mapea o asigna a una clave un determinado valor. Un mapa no puede contener elementos duplicados, ya que para un mismo elemento con una determinada clave, éste siempre tendrá que devolver el mismo valor.

Similar a esta interfaz es la interfaz SortedMap, solamente se diferencian en que los elementos del mapa están ordenados, haciendo por tanto en principio que las consultas o búsquedas sean más rápidas y eficientes que si no estuvieran ordenados.

La interfaz Set es la interfaz que nos definirá un conjunto de elementos. También al igual que la anterior no admite elementos duplicados, ni nulos. Y al igual que la anterior también tenemos la versión SortedSet que es un conjunto de elementos, pero ordenados, con las ventajas que aporta la ordenación que hemos comentado anteriormente.

CONCLUSIONES SOBRE LAS INTERFACES DEL PAQUETE JAVA.UTIL

Como conclusión a esta revisión de las principales interfaces del paquete java.util, diremos que se trata de un conjunto de interfaces que nos van a permitir hacer programas algo más avanzados, ya que nos facilitan por ejemplo el tener un conjunto de elementos ordenados, una lista de usuarios donde podemos consultar sus nombres por ejemplo, o un mapa donde a cada usuario lo podemos localizar por una clave de identificación.

BREVE DESCRIPCIÓN DE LAS PRINCIPALES CLASES EN EL PAQUETE JAVA.UTIL

A continuación describiremos brevemente las principales clases que hemos seleccionado del paquete java.util. Al igual que las interfaces veremos más adelante detalladamente con ejercicios y ejemplos resueltos cada una de ellas.

CALENDAR

Es una clase abstracta que nos obliga a implementar diversos métodos para crear un calendario, pero normalmente se utiliza la subclase de esta `GregorianCalendar` para utilizar un calendario gregoriano, que es el calendario de uso más habitual en la mayoría de países.

DATE

Es una clase que permite representar un instante específico en el tiempo con precisión de milisegundos. En las versiones más recientes de Java muchos de los métodos de esta clase están "deprecated", es decir, obsoletos y de uso no recomendado. Para reemplazar funcionalidades de esta clase aparece el paquete `java.time`. Este paquete proporciona funcionalidades sobre fechas y calendarios, y es el que debemos usar para el manejo de fechas y tiempo. ¿Por qué se menciona aquí la clase `Date`? Porque ha sido una clase muy usada en el pasado y nos podemos encontrar con código que usa versiones anteriores o que sigue usando la clase `Date` del paquete `java.util`.

Nosotros no usaremos esta clase, sino las clases proporcionadas dentro del paquete `java.time`.

HASHMAP, HASHSET, TREEMAP Y TREESET

Son las principales clases que respectivamente implementan las interfaces para `Map`, `Set`, `SortedMap` y `SortedSet` respectivamente.

LINKEDLIST

Es la clase que principalmente implementa la interfaz `List` junto quizás con `ArrayList`. Es doblemente enlazada lo que permite el avance por una lista tanto en un sentido como en otro, acceder al primer elemento y al último elemento directamente o la inclusión de elementos nulos.

STRINGTOKENIZER

Es una de las clases más utilizadas dentro del paquete `java.util`. Esta clase permite partir un string en tokens (un token es una porción de un string más grande al cual le decimos por qué carácter debemos de partirlo. Por ejemplo si tenemos el string "Un coche rápido y rojo" y deseamos obtener los tokens separados por un espacio en blanco, tendríamos como resultado 5 tokens que son: "Un", "coche", "rápido", "y", "rojo").

CONCLUSIONES SOBRE LAS CLASES DEL PAQUETE JAVA.UTIL

Como conclusión a este apartado, indicar que el paquete `java.util` proporciona un conjunto de clases muy interesantes y de gran potencialidad. Cuando se programa mucho y se tienen conocimientos de

varios lenguajes, nos damos cuenta de la gran herramienta que son las bibliotecas Java como java.util, que facilitan enormemente el desarrollo de código a los programadores. Disponer de buenas bibliotecas ha sido uno de los factores que ha contribuido al éxito mundial del lenguaje Java.

Próxima entrega: CU00917C

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=58&Itemid=180