Unidad 1 - Reto de Programación

Colecciones en Java - ¿Cómo almacenar datos en memoria?

La idea es muy simple, cuando queremos trabajar con un conjunto de elementos, necesitamos un almacén donde poder guardarlos.

En Java, se emplea la interface genérica **Collection** para este propósito. Gracias a esta interface, podemos almacenar cualquier tipo de objeto y podemos usar una serie de métodos comunes, como pueden ser: añadir, eliminar, obtener el tamaño de la colección… ya que se trata de métodos definidos por la interface que obligatoriamente han de implementar las subinterfaces o clases que hereden de ella.

```
Llamamos Colección a un grupo de objetos individuales que se representan como una única
```

Punto de partida – Recursos en Internet

Existen multitud de páginas con código de ejemplo para el uso de las colecciones en Java. A la hora de buscar debéis ser cuidadosos, porque hay sitios con código antiguo y obsoleto, por lo que tenéis que centrar la búsqueda en páginas actualizadas recientemente.

Un buen ejemplo, con gran cantidad de código de ejemplo y con una explicación detallada del framework Collection de Java es el siguiente:

- GeeksForGeeks

Aunque está en inglés (hay que ir acostumbrándose a leer documentación técnica en la lengua de Shakespeare), puedes encontrar otros recursos interesantes en castellano.

Este artículo no ofrece únicamente una visión global de las colecciones y las clases que componen el framework, sino que al final del artículo podrás ver una buena lista de enlaces bajo el epígrafe *What You Should Learn in Java Collections?* que te ayudarán a completar este reto.

Programación funcional – Java API Stream

Los streams (de la API Stream, no confundir con los relacionados con ficheros) nos dan la posibilidad de realizar operaciones funcionales sobre los elementos de las colecciones, siguiendo la filosofía de la denominada “programación funcional”.

En este artículo tienes una breve introducción al uso del API Stream en Java:

- aprenderaprogramar.com - Uso de API Stream

Expresiones lambda – Java API Stream

Las expresiones lambda te permiten reflejar de forma compacta clases o interfaces simples que contienen un único método.

En este caso vamos a echar mano de la documentación oficial de Oracle para tener una primera aproximación a las expresiones lambda y de otro artículo para ver una aplicación más concreta a las colecciones y los streams.

- Oracle - Expresiones lambda
- Javatpoint - Expresiones lambda

No es un tipo de programación trivial, aunque sí es bastante efectiva y rápida una vez que se controla. En este reto sólo pretendo que busquéis cómo hacer determinadas acciones con el uso de expresiones lambda, partiendo de ejemplos que encontréis.

Reto: Usar colecciones con soltura

En este reto os voy a proponer que escojáis una de las dos colecciones más usadas e investiguéis cómo usarlas en Java.

ArrayList HashMap

En AULES compartiré las mejores soluciones para cada una de las colecciones.

¿En qué consiste entonces el reto?

Pues tenéis que realizar un breve manual técnico, que os servirá de referencia para las actividades que desarrolléis a lo largo del curso.
Complementadlo con ejemplos de código y descripciones sencillas de la funcionalidad de ese código.

Intenta que la colección contenga un objeto, mejor que tipos primitivos o sus clases Wrapper (envolturas). Por ejemplo un objeto persona con nombre, apellidos, edad, altura, peso, sexo, etc.

Esta guía debe contener, al menos los siguientes elementos

a) Definición y creación de una colección
◦ Diferentes constructores que consideréis útiles

b) Poner ejemplos de métodos / propiedas generales
◦ tamaño

◦ Acceso por índice / clave
◦ ...

c) Añadir datos a la colección
◦ Añadir elementos desde el constructor.
◦ Añadir elementos a partir de otras colecciones.
◦ Añadir elementos mediante código.
◦ Eliminar elementos mediante código.

d) Recorrer la colección
◦ Con un bucle for
◦ Con un bucle foreach de Java
◦ Usando Iterator
◦ Con expresiones lambda
Aquí hay un caso especial, que es cuando se quiere modificar (añadir/eliminar/actualizar) la colección mientras se está recorriendo. En este caso no todas las formas de recorrerla son válidas. Investígalo.

e) Búsqueda de elementos por clave/valor o por alguna propiedad del objeto
◦ Con un bucle (for / foreach / Iterator)
◦ Con expresiones lambda
◦ Con API Stream

f) Obtención de subcolecciones
◦ Con un bucle (for / foreach / Iterator)
◦ Con expresiones lambda
◦ Con API Stream (filtrado / collect)

g) Ordenación de elementos
◦ Con funciones de Collection
◦ Con expresiones lambda
◦ Con API Stream

h) Cualquier otro aspecto que quieras añadir

La guía debe ser un documento “vivo” por lo que en cualquier momento del curso podéis modificarla y completarla con nuevos apartados o ejemplos.

Recursos adicionales

Hay un site muy interesante con recursos muy buenos a la hora de saber cómo hacer algo concreto en Java. Os dejo el enlace a los artículos relacionados con Colecciones.

- Baeldung

Rúbrica de evaluación

Reto 1 - Colecciones Java Listo para su uso

Apartados completados	Todos los apartados están completos. *2 puntos*	La mayoría de los apartados están completos, pero falta alguno o algunos. *1 puntos*	No se han completado la mayoría de los apartados. *0 puntos*
Corrección de los ejemplos aportados	Los ejemplos propuestos son correctos y/o adecuados. *5 puntos*	La gran mayoría de los ejemplos están bien, pero hay algunos que tienen errores importantes. *3 puntos*	Los ejemplos no son correctos. *0 puntos*
Uso de API Stream	La parte de API Stream se ha tratado correctamente en todos los apartados indicados. *4 puntos*	La parte de API Stream se ha usado, pero no en todos los apartados solicitados. *2 puntos*	No se ha trabajado la parte de API Stream en los apartados indicados. *0 puntos*
Uso de expresiones lambda	La parte de expresiones lambda se ha tratado correctamente en todos los apartados indicados. *4 puntos*	La parte de expresiones lambda se ha usado, pero no en todos los apartados solicitados. *2 puntos*	No se ha trabajado la parte de expresiones lamdba en los apartados indicados. *0 puntos*
Formato de la documentación (Estructura: título, índice, división apartados, formato, bibliografía)	La documentación presentada tiene una estructura adecuada y una presentación profesional. *5 puntos*	La documentación no se ajusta exactamente a lo indicado y le falta mejorar estructura y/o presentación. *3 puntos*	La documentación tiene un formato poco profesional y no se ajusta a lo indicado. *0 puntos*