

Ejercicios adicionales UT6 –

Colecciones de tamaño flexible: HashSet, HashMap.

Ejercicio de repaso.

Abre el proyecto que se te ha proporcionado. Lee su código y complétalo. La clase **DemoFormacion** te servirá para probar las otras dos.

La clase **Cursillo** no hay que modificarla. Representa a un cursillo con su nombre y horas de duración. Lee el código de esta clase.

La clase **Formacion** guarda la relación de cursos realizados por una serie de alumnos. Lo hace a través de un atributo definido de la forma:

```
private TreeMap<String, ArrayList<Cursillo>> mapCursos;
```

La clave en el *map* representa el nombre del alumno (siempre se guarda en mayúsculas) y el valor asociado es la lista de cursillos realizados por ese alumno.

Completa en esta clase:

- a) el constructor – inicializa adecuadamente el map
- b) **public void addCursillo(String alumno, String curso, int horas)** – dado un nombre de alumno se añade un nuevo cursillo a su lista de cursos. Si la clave no existe se crea, si ya existe únicamente se añade el cursillo a su lista de cursos. Si el cursillo existe ya no se añade (para hacer esto únicamente tendrás que utilizar el método `contains()` de `ArrayList`. Todo funcionará bien porque se ha incluido en la clase `Cursillo` un método `equals()`). La clave siempre se añade en mayúsculas
- c) **public int totalHorasCursillosDe(String alumno)** – dado un alumno obtener el total horas de cursillos realizadas por él. Si el alumno no existe se devuelve -1
- d) **public String alumnoConMasHorasDeCursillos()** – devuelve el nombre del alumno con más horas de cursillos realizadas
- e) **public void alumnosTotalCursillos()** – Muestra cada alumno con el nº total de cursillos realizados (usando `keySet`)
- f) **public void alumnosTotalHorasCursillos()** – Muestra cada alumno con el total de horas de cursillos realizadas. Usar `Map.Entry`.
- g) **public String toString()** – Representación textual del *map* mostrando cada alumno y los nombres de los cursillos realizados - Usando `Map.Entry` y un iterador para recorrer el conjunto de entradas

Añade las clases **Formacion** y **Cursillo** al paquete *formacion.modelo* y la clase **DemoFormacion** al paquete *formacion.demo*. Haz las modificaciones adecuadas para que todo funcione correctamente.

Desde el IDE:

- Crea un *jar* de nombre *formacion.jar* listo para ser distribuido y ejecutado
- Ejecuta el *jar* desde línea de comandos para comprobar que funciona (anota el comando que has utilizado)
- borrar el *jar* anterior

Sal a línea de comandos:

- Sitúate en el directorio base del proyecto (*REPA Alumnos y cursillos AL*)
- Crea de nuevo el *jar* ejecutable *formacion.jar* (anota el comando que has utilizado) añadiendo únicamente los ficheros *.class* al *jar*
- Visualiza el contenido del *jar* anterior (anota el comando)
- Ejecuta el *jar* anterior para comprobar que funciona (anota el comando)

Possible ejecución

Después de añadir

ALBERTO

Cursos: XML

ANA

Cursos: PHP JAVA C#

JOSE

Cursos: JAVA JOOMLA

LUIS

Cursos: SEO

PEPE

Cursos: PHP JAVA

Total horas de cursillos realizadas por PEPE = 110

Total horas de cursillos realizadas por ANA = 180

Total horas de cursillos realizadas por JESUS = -1

Con más horas de cursillos: ANA

Alumnos y total cursillos

ALBERTO Total cursos realizados 1

ANA Total cursos realizados 3

JOSE Total cursos realizados 2

LUIS Total cursos realizados 1

PEPE Total cursos realizados 2

ALBERTO Horas cursillos realizadas: 70

ANA Horas cursillos realizadas: 180

JOSE Horas cursillos realizadas: 50

LUIS Horas cursillos realizadas: 10

PEPE Horas cursillos realizadas: 110