

Ejercicios adicionales UT6 (VI) –

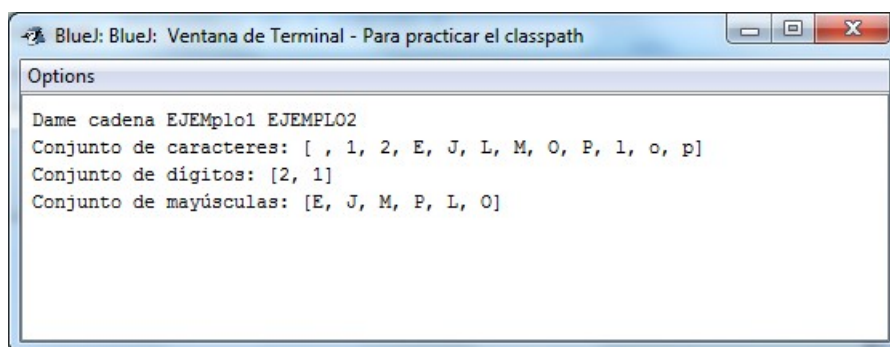
Colecciones de tamaño flexible: HashSet, HashMap.

Ejercicio 11. En este ejercicio vas a practicar con el *classpath*. Hazlo primero en BlueJ y luego en IntelliJ. Lee previamente la definición en las diapositivas y lee también lo incluido acerca del *classpath* en Moodle.

- descarga el fichero *utils.jar* y sitúalo en el directorio [C:\MisLibrerias](#)
- desde línea de comandos sitúate en [C:\MisLibrerias](#) y visualiza el contenido de ese fichero *jar*
 - anota el comando con el que lo has visualizado
 - verás que contiene una clase compilada *UtilsString.class*. En qué paquete está la clase? Anótalo.
 - El interface de la clase es el siguiente

Method Summary	
<code>java.util.TreeSet<java.lang.Character></code>	getConjuntoCaracteres (java.lang.String cadena) Dada una cadena obtiene y devuelve su conjunto de caracteres Importa el orden en el conjunto
<code>java.util.HashSet<java.lang.Character></code>	getConjuntoDigitos (java.lang.String cadena) Dada una cadena obtiene su conjunto de caracteres numéricos No importa el orden en el conjunto Usa algún método de la clase Character para comprobar si un caracter es numérico o no
<code>java.util.LinkedHashSet<java.lang.Character></code>	getConjuntoMayusculas (java.lang.String cadena) Dada una cadena obtiene su conjunto de letras mayúsculas en el orden en el que aparecen en la cadena

- Descarga ahora el proyecto *Para practicar el classpath AL* en tu directorio de trabajo habitual, por ej, *C:\Programación\UT6* y ábrelo en BlueJ
- Completa el método *main()* de forma que muestres el siguiente resultado (solo tendrás que llamar adecuadamente a los métodos anteriores y mostrar el resultado en pantalla). Ten en cuenta que la clase *UtilsString* no está en tu proyecto sino compilada dentro del fichero *utils.jar*. Haz las modificaciones necesarias tanto en la clase *UsandoClassPath* como en *BlueJ*. Ejecuta desde *BlueJ* y verifica que funciona.



- Sal a línea de comandos y sitúate en el directorio base del proyecto *Para practicar el classpath AL*.
- Visualiza el *classpath* en este momento. Verás que no incluye la ruta [C:\MisLibrerias](#)
- Ejecuta ahora la aplicación teniendo en cuenta que la clase *UsandoClassPath* hace uso de *UtilsString* que está en *utils.jar*
- Anota el comando que has ejecutado en el apartado anterior
- Ahora crea un proyecto IntelliJ *Para practicar el classpath AL*, incluye la clase *UsandoClassPath* y haz el apartado d)

Ejercicio 12.

Descarga el proyecto IntelliJ *Otro conversor romanos AL*. Solo incluye una clase `ConversorRomanos` con un método `int aArabigo(String romano)` similar al que has hecho en el ejercicio 10.

Observa que el atributo `map` está definido como `LinkedHashMap`. Qué caracteriza a esta colección?

Ejecuta el método y verifica que funciona (seguimos con la notación romana antigua).

Completa el método `String aRomano(int arabigo)`. Este método convierte el n° arábigo recibido como parámetro (suponemos menor o igual a 4000, asumimos esto) a n° romano en notación antigua. Por ej. 1573 es “MDLXXIII” - 345 es “CCCXXXV” - 23 es XXIII

Una vez compruebes que funciona todo incluye la clase en el paquete *otroconversor.romanos*. Compila todo.

Exporta la clase `ConversorRomanos.class` al fichero *otroconversor.jar* y copia esta librería en [C:\MisLibrerías](#)

Crea ahora un nuevo proyecto *Probando otro Conversor* y añade una clase solo con el método `main()` `AppOtroConversor`. Dentro de esta clase pide al usuario varios números romanos y muestra sus correspondientes arábigos y viceversa. Tendrás que utilizar la clase `ConversorRomanos` que está compilada en la librería *otroconversor.jar*.

```
Dame n° arábigo: 1573
Su romano es: MDLXXIII
Dame n° arábigo: 345
Su romano es: CCCXXXV
Dame n° arábigo: 24
Su romano es: XXIII
Dame n° arábigo: 129
Su romano es: CXXVIII
-----
Dame n° romano: MDLII
Su arábigo es: 1552
Dame n° romano: CII
Su arábigo es: 102
Dame n° romano: XIII
Su arábigo es: 14
Dame n° romano: DCLIII
Su arábigo es: 653
```