



- Utilizando el método genérico del apartado anterior, aproximar la probabilidad de obtener un resultado total de siete al lanzar dos dados.

### Ejercicio 7.2 (0.6 puntos)

Supongamos un archivo que contiene, en formato texto, información sobre el número de alumnos matriculados en cada grupo de cada asignatura de una cierta titulación. Esta información viene dada con el siguiente formato:

```
FUNDAMENTOS DE COMPUTADORES (Código 803265 - Grupo A): 58 matriculados.  
FUNDAMENTOS DE ELECTRICIDAD Y ELECTRÓNICA (Código 803261 - Grupo A): 74 matriculados.  
FUNDAMENTOS DE COMPUTADORES (Código 803265 - Grupo B): 61 matriculados.  
GESTIÓN EMPRESARIAL (Código 803260 - Grupo A): 56 matriculados.  
GESTIÓN EMPRESARIAL (Código 803260 - Grupo B): 64 matriculados.  
MATEMÁTICA DISCRETA Y LÓGICA MATEMÁTICA (Código 803263 - Grupo C): 90 matriculados.  
...
```

Utilizando expresiones regulares se pide obtener, para cada línea, el código de la asignatura y el número de matriculados. Esta información se guardará en un `TreeMap<Integer, Integer>` que asocie cada código de asignatura con el número de matriculados en la misma. Pueden existir distintas líneas no consecutivas con información sobre la misma asignatura, como ocurre en el ejemplo mostrado anteriormente. En ese caso, el programa ha de guardar una única entrada en el `TreeMap` conteniendo el número total de matriculados en esta asignatura. En nuestro ejemplo, el `TreeMap` ha de asociar el código 803265 con el número total de matriculados 119 (= 58 + 61), y el código 803260 con el número 120 (= 56 + 64).

### Ejercicio 7.3 (0.4 puntos)

Modificar la clase `ScannerCasero` del Ejercicio 3.2 para que implemente la interfaz `Iterable<Integer>`. De este modo, el código de ejemplo mostrado en dicho ejercicio puede transformarse en la siguiente versión más simple.

```
public static void main(String[] args) {  
    ScannerCasero scc = new ScannerCasero("34:56:11:546:1");  
    for (Integer i : scc) {  
        System.out.println(i);  
    }  
}
```