



APELLIDOS, Nombre

GRUPO

MÁQUINA

Se desea crear una aplicación que permita asignar cargos electos a diversos partidos en unas elecciones, considerando el sistema conocido como ley de D'Hondt. El sistema de D'Hondt es una fórmula electoral, creada por Victor d'Hondt, que permite obtener el número de cargos electos asignados a las candidaturas, en proporción a los votos conseguidos. El procedimiento consiste en, tras escrutar todos los votos, calcular una serie de divisores para cada partido (o lista) político. La fórmula de los divisores es V/N , donde V representa el número total de votos recibidos por el partido, y N representa cada uno de los números enteros de 1 hasta el número de cargos electos de la circunscripción objeto de escrutinio. Una vez realizadas las divisiones de los votos de cada candidatura por cada uno de los divisores desde 1 hasta N , la asignación de cargos electos se hace ordenando los cocientes de las divisiones de mayor a menor y asignando a cada uno un escaño hasta que éstos se agoten.

- 1) **(1.5 ptos.)** La clase `Divisor` debe mantener información sobre un partido (`String`) y una cantidad de votos (`int`), representando divisores como los mencionados anteriormente. La clase debe incluir elementos que permitan:
 - a. **(0.25 ptos.)** Construir objetos de la clase a partir del nombre de un partido y una cantidad de votos, proporcionados como argumentos. Obtener el nombre del partido a través de un método `String getPartido()`. Obtener una representación textual de los objetos de la forma: `partido --> votos`.
 - b. **(0.50 ptos.)** Dos objetos de la clase `Divisor` son iguales si coinciden el número de votos y el nombre del partido, independientemente de que esté en mayúsculas o minúsculas.
 - c. **(0.75 ptos.)** Un objeto de la clase `Divisor` es menor que otro cuando el número de votos del primero es mayor que el número de votos del segundo. En caso de que el número de votos coincida en ambos, se considerará menor el del partido que alfabéticamente sea menor.
- 2) **(2.5 ptos.)** Defínase una clase `EleccionDHondt` que almacene en una correspondencia (`Map`) el número de votos conseguidos por diversos partidos en unas elecciones. Este atributo debe ser `protected`, y la clase incluirá:
 - a. **(0.5 ptos.)** Un constructor con un argumento de tipo `Map<String,Integer>` conteniendo los votos asociados a cada partido.
 - b. **(2.0 ptos.)** Un método `Map<String,Integer> reparto(int nroCargos)` que devuelve la asignación de cargos electos que le corresponde a cada partido según la Ley D'Hondt, teniendo en cuenta que el número total de cargos a distribuir se proporciona como argumento. El algoritmo para realizar el reparto consistirá en crear un conjunto ordenado de divisores (objetos de la clase `Divisor`) para todos los partidos, dividiendo el número de votos obtenidos en cada caso por los enteros 1 hasta el número total de cargos. A continuación, bastará seleccionar los primeros `nroCargos` elementos de este conjunto ordenado y devolver una aplicación que asocie a cada partido el número de objetos divisores seleccionados que le correspondan. Por ejemplo, para una distribución de votos como la siguiente:

	Partido A	Partido B	Partido C	Partido D	Partido E
Votos	340.000	280.000	160.000	60.000	15.000

Y un número de cargos electos a distribuir de 7, el algoritmo debería construir objetos `Divisor` que se correspondan con los elementos de la siguiente tabla:

	1	2	3	4	5	6	7
A	340.000	170.000	113.333	85.000	68.000	56.667	48.571
B	280.000	140.000	93.333	70.000	56.000	46.667	40.000
C	160.000	80.000	53.333	40.000	32.000	26.667	22.857
D	60.000	30.000	20.000	15.000	12.000	10.000	8.571
E	15.000	7.500	5.000	3.750	3.000	2.500	2.143

Las celdas sombreadas representarían los 7 divisores más pequeños según el orden natural definido en la clase `Divisor`. Por lo tanto, el método devolvería finalmente una aplicación con la información siguiente:

	Partido A	Partido B	Partido C
Cargos	3	3	1

- 3) **(2 ptos.)** En España (y otros muchos países) no se utiliza la ley D'Hondt directamente, sino una modificación consistente en excluir del reparto a los partidos que no superen un mínimo porcentaje de votos sobre el total. Defínase una clase `EleccionDHondtModificado` que en su constructor, además de la correspondencia con los votos asociados a cada partido, incluya el porcentaje (`double`) mínimo para participar en el reparto.
- 4) **(4 ptos.)** Las elecciones municipales en España siguen la ley D'Hondt modificada, en la que el porcentaje mínimo sobre el total de votos emitidos ha de ser del 5% (en el caso de las elecciones al Congreso de los Diputados, el porcentaje mínimo es del 3%). Defínase una clase `EleccionesMunicipales` que almacene en estructuras adecuadas la siguiente información sobre las elecciones realizadas en diversos municipios: i) por cada municipio (`String`), los votos recibidos por cada uno de los partidos; ii) para cada municipio, el número de concejales a cubrir; y iii) los resultados electorales, organizados por cada municipio, asociando a cada partido el número de concejales obtenidos. La clase debe incluir también:
 - a. **(0.5 ptos.)** Un constructor que tenga como argumento el nombre de un fichero con información sobre los resultados de las elecciones. El fichero vendrá organizado por líneas, de forma que cada una de ellas incluirá información sobre la ciudad, el número total de concejales que le corresponde a la ciudad, y los votos recibidos por cada uno de los partidos. Por ejemplo, el fichero `"eleccionesAndalucia.txt"` muestra esta información correspondiente a las ocho capitales andaluzas en las pasadas elecciones municipales, donde una de las líneas contiene:

```
Málaga:31:PP    123655;PSOE-A  57245;IULV-CA  25354; ...
```

(Nota: El separador entre el partido y el número de votos es un tabulador: `\t`. Cualquier error en el formato del fichero deberá contemplarse con el lanzamiento de una excepción no comprobada `EleccionesException`, incluyendo un mensaje descriptivo del error.)

El constructor debe realizar las acciones necesarias para incorporar la información incluida en el fichero (ver apartado (b)), y realizar el reparto con el porcentaje mínimo del 5% (ver apartado (c)).
 - b. **(2,0 ptos.)** Un método `void leerVotos(String)` para obtener la información de unas elecciones a partir de un fichero, organizado tal y como se indica en el apartado anterior:
 - c. **(0.5 ptos.)** Un método `void reparto(double)`, que pueda rehacer el reparto de concejales, considerando como porcentaje mínimo para participar en la asignación el que se pase como argumento.

- d. (1.0 ptos.) Dos métodos para guardar la asignación de concejales en un fichero de texto o un flujo de salida, respectivamente: `void print(String)` y `void print(PrintWriter)`. La salida debe tener el formato que se puede observar en el fichero “resultados.txt”.

La siguiente clase principal:

```
import dhondt.*;

public class PruebaElecciones {
    public static void main(String args[]) {
        try {
            EleccionesMunicipales em =
                new EleccionesMunicipales("eleccionesAndalucia.txt");
            em.print("resultados.txt");
            em.print(new PrintWriter(System.out));
        } catch (IOException ioe) {
            System.err.println("Error de E/S: " + ioe.getMessage());
        } catch (EleccionesException ee) {
            System.err.println(ee.getMessage());
        }
    }
}
```

muestra el funcionamiento de las clases anteriormente descritas. Consúltense los ficheros de prueba de texto proporcionados.

PROGRAMACIÓN ORIENTADA A OBJETOS

NOTAS PARA LA REALIZACIÓN DEL EJERCICIO

(Léase detenidamente antes de comenzar el ejercicio)

El ejercicio se almacenará en el directorio **C:\POO**. En caso de que no exista deberá crearse, y si ya existiese, deberá borrarse todo su contenido antes de comenzar.

CADA FICHERO DEBERÁ IDENTIFICARSE CON EL NOMBRE DEL ALUMNO, TITULACIÓN, GRUPO Y NÚMERO DE EQUIPO.

La evaluación tendrá en cuenta la claridad de los algoritmos, del código y la correcta elección de las estructuras de datos, así como los criterios de diseño que favorezcan la reutilización.

ESTÁ PERMITIDO:

- Consultar el API que se encuentra en el disco de red.

NO ESTÁ PERMITIDO:

- Utilizar otra documentación electrónica o impresa.
- Intercambiar documentación con otros compañeros.
- Utilizar soportes de almacenamiento.

IMPORTANTE: APAGAD LOS DISPOSITIVOS ELECTRÓNICOS DE COMUNICACIÓN

Una vez terminado el ejercicio subir un fichero comprimido (.jar, .zip o .rar) sólo con los **fuentes** que hayáis realizado a la tarea creada en el campus virtual para ello.