

Técnicas Actuales de Programación 2022/2023

Examen de Convocatoria Extraordinaria – 27 de Junio

Una empresa que ofrece servicios relacionados con la sostenibilidad desea crear un software que permita comparar distintas especies de árboles en base a sus características. Un individuo o árbol cualquiera puede ser caracterizado mediante los siguientes atributos:

- Nombre de la especie
- Diámetro del tronco
- Altura
- Almacenamiento de carbono
- Retención de ozono (O₃)
- Almacenamiento de CO₂

La empresa cuenta con un fichero de datos en formato de texto compuesto por campos separados por tabuladores con la siguiente estructura:

ESPECIE	DIAM	ALT	ACARBON	RO3	ACO2
Abies alba	42,0168	12	334,03	337,359	9,70687
Abies alba	40,107	19	307,04	348,699	9,25328
Abies alba	23,2366	13	100,83	116,233	5,26167
Abies pinsapo	30,876	12	207,02	189,942	5,76062
Abies pinsapo	39,152	12	335,2	297,67	7,34819

Cada fila (excepto la primera) hace referencia a un individuo (árbol) y cada columna indica el valor de uno de los atributos previamente descritos. En la siguiente dirección puede encontrar un fichero de texto (codificación UTF8) de muestra: <https://gtts.ehu.es/German/Docencia/2223/TAP/examenes/trees.txt>.

1 – Clase Tree

Cree la clase **Tree** que almacenará la información promedio de los atributos de una especie:

- diameter → diámetro del tronco
- height → altura
- carbonStorage → almacenamiento de carbono
- o3Retention → retención de ozono (O₃)
- co2Storage → almacenamiento de CO₂

La clase contará además con los siguientes métodos públicos (puede hacer los privados que considere necesarios):

- **public Tree(String name)**
Constructor que inicializará el objeto en base a su nombre de especie. Inicialmente, al no contar con ningún individuo, se entenderá que el valor promedio de todos los atributos es 0.
- **public string getName()**
Devuelve el nombre de especie
- **public void addSingleTreeSample(double diameter, double height, double carbonStorage, double O3Retention, double co2Storage)**
Añade los atributos de un nuevo individuo (las mediciones relativas a un único árbol) a la especie.

- **public double getAttribute(??? attribute)**
Devuelve el valor promedio de alguno de los atributos (diámetro, altura, carbono, etc.):
- **public double getDistanceFor(Tree other, ??? attribute)**
Devuelve cuál es la distancia (valor absoluto de la diferencia) frente a otra especie en base a alguno de los atributos (diámetro, altura, carbono, etc.).
- **public double toString()**
Muestra el valor promedio de los atributos de la especie.

2 – Clase TreeDB

Cree una clase de **TreeDB** que almacene la información de un conjunto de especies de árboles. La clase contendrá los siguientes métodos públicos (puede hacer los privados que considere necesarios):

- **public TreeDB(String fileName, String encoding)**
Constructor. Inicializa la BD a partir de un fichero de texto con el formato indicado previamente.
- **public Set<String> getAllNames()**
Devuelve el conjunto de nombres de especies de la base de datos.
- **public Tree get(String name)**
Devuelve el Tree correspondiente a un nombre de especie, o null si no existe.
- **public List<Tree> search(String s)**
Devuelve la lista de Trees cuyo nombre de especie contengan la cadena de caracteres s (case-insensitive)
- **public String mostSimilar(String speciesName, ??? attribute)**
Devuelve el nombre de especie del Tree más similar (de menor distancia euclídea) en base a alguno de los atributos (diámetro, altura, carbono, etc.).
- **public static void main(String[] args)**
Este “main” permitirá comprobar que la recogida de datos se hace correctamente, mostrando la lista de todas las especies contenidas en el fichero de muestra (el formato es orientativo):

```

Abies alba: (DIAMETER: 35,120) (HEIGHT: 14,667) (CARBONSTORAGE: 247,300) (O3RETENTION: 267,430) (CO2STORAGE: 8,074)
Abies pinsapo: (DIAMETER: 35,014) (HEIGHT: 12,000) (CARBONSTORAGE: 271,110) (O3RETENTION: 243,806) (CO2STORAGE: 6,554)
Acer campestre: (DIAMETER: 9,372) (HEIGHT: 4,278) (CARBONSTORAGE: 26,623) (O3RETENTION: 32,444) (CO2STORAGE: 2,046)
Acer monspessulanum: (DIAMETER: 14,483) (HEIGHT: 5,250) (CARBONSTORAGE: 92,825) (O3RETENTION: 42,524) (CO2STORAGE: 9,548)
Acer negundo: (DIAMETER: 19,326) (HEIGHT: 7,453) (CARBONSTORAGE: 95,945) (O3RETENTION: 115,503) (CO2STORAGE: 7,474)
Acer opalus: (DIAMETER: 3,501) (HEIGHT: 3,375) (CARBONSTORAGE: 4,211) (O3RETENTION: 6,733) (CO2STORAGE: 1,695)
Acer platanoides: (DIAMETER: 11,762) (HEIGHT: 5,929) (CARBONSTORAGE: 62,337) (O3RETENTION: 44,414) (CO2STORAGE: 5,620)
...

```

3 – Aplicación gráfica TreeInfo

La aplicación construirá un objeto TreeDB del apartado anterior a partir del fichero de muestra y facilitará el análisis de la información. La aplicación gráfica constará de tres áreas con información:

1. Zona “Especies disponibles”: inicialmente mostrará toda la lista de especies disponibles, que se irá reduciendo en función del texto que se introduzca en un campo de texto, mostrando sólo aquellas que lo contengan en su nombre (case-insensitive). Esto puede hacerse de diversas maneras: cada

vez que se pulsa una tecla, o al pulsar “enter” tras escribir en el campo, o añadiendo un botón para llevar a cabo la acción, etc.

2. Zona “valores de los atributos”: Cuando el listado en la primera zona se limite a una única especie, en esta zona se mostrarán sus atributos.
3. Zona “Más próximas por atributo”: Cuando el listado en la primera zona se limite a una única especie, en esta zona se mostrarán las especies más similares frente a ella en base a cada uno de los atributos (diámetro, altura, carbono, etc.) indicando el valor de dicho atributo y la distancia del mismo con el de la especie seleccionada.

La aplicación podría tener la siguiente apariencia (esto es orientativo):

