**(Los \\ es para hacer salto de línea en latex, no te preocupes por eso xD)**

En un primer lugar, resolvemos el problema que se da en el enunciado de la práctica para poder utilizarlo de base.\\

Para la lectura del archivo, creamos una clase Reader, en la cual implementamos la lectura de los archivos de entrada.\\

Para la construcción del árbol final, implementamos una clase Node, cuyo constructor recibe un String, que será el valor seleccionado tras aplicar el ID3, y otro nodo para seguir en una futura recursión.\\

En la clase ID3 es donde implentamos el algoritmo como tal. Tenemos dos listas, una de String attributes, donde almacenamos los atributos leídos, y data, que es una de listas de string que almacena los valores de los atributos. Tenemos un atributo string targetAttribute, el cual va a ser el valor del primer nodo. Por defecto, elegimos el primer atributo, pero este puede cambiar en la primera iteracción de la recursión. A partir de la primera recursión, el valor de este atributo no cambia, ya que va a ser el nodo del cual cuelgue el resto del árbol. Guardamos el resto de atributos no elegidos en una lista de string llamada noTargetAttributes, y a partir de aquí aplicamos el ID3.\\

Para empezar guardamos en un map llamado targetValues, el número de veces que aparece cada valor del atributo targetAttribute. Si el tamaño del map es de 1, significa que hemos llegado al final de ese subárbol. Si por otro lado, está vacía, no tenemos información por lo que no seguimos por esa rama, y volvemos hacia atrás.\\

Si todo va bien, calculamos la entropía del targetValue con la función getEntropy, la cual cálcula la entropía según la fórmula dada en las diapositivas de teoría. Para ello utilizamos un bucle for, el cual cálcula.