

Índice

| | |
|----------------|---|
| Tipo test..... | 2 |
|----------------|---|

Tipo test

1. Con la clase `DataInputStream` leeremos:

Objetos de tipo primitivo de java como números enteros, decimales, caracteres, booleanos, etc.

2. ¿Cuál de las siguientes clases en Java se utiliza para leer caracteres y si es necesario, devolverlos al flujo de entrada para una lectura posterior?

`PushbackReader`

3. ¿Qué método de la clase `DataOutputStream` se usa para escribir un objeto de tipo `Long`?

`writeLong()`

4. Para instanciar un nuevo hilo de ejecución usaremos la clase...

`Thread`

5. Con el método `readLine()` de la clase `LineNumberReader`

Leeremos el contenido de una línea

6. ¿Qué método nos indica que un token es de tipo palabra?

`TT_WORD`

7. Que nos indica el método `TT_EOF` de la clase `StreamTokenizer`

Fin del fichero

8. ¿Qué clase en Java permite la lectura y escritura simultánea entre dos hilos del mismo proceso?

`PipedOutputStream` y `PipedInputStream`

9. ¿Qué conocido método devuelve el byte al stream de datos?

`Unread()`

10. ¿Qué clase en Java se utiliza para leer una secuencia de caracteres desde un array de caracteres?

`CharArrayReader`

11. La clase `LineNumberReader` tiene un método que nos indica la línea en la que estamos posicionados leyendo. ¿De qué número hablamos?

`getLineNumber()`

12. Para la ejecución de una tubería de lectura y otra de escritura:

Se creará un proceso con 2 hilos independientes

13. ¿Qué clase se utiliza para analizar un fichero por "trozos" o "fragmentos" en Java?

StreamTokenizer

14. StreamTokenizer dispone de algunos métodos estáticos que nos dan información de la tipología de los distintos Tokens:

TT_EOF, TT_EOL, TT_WORD, TT_NUMBER

15. ¿Qué representa el atributo sval en la clase StreamTokenizer?

Almacena el valor del token cuando es una palabra

16. ¿Qué se crea utilizando PipedOutputStream?

Una tubería de salida