Class Text

This class stores text using standard UTF8 encoding. It provides methods to serialize, deserialize, and compare texts at byte level. The type of length is integer and is serialized using zero-compressed format.

In addition, it provides methods for string traversal without converting the byte array to a string.

Also includes utilities for serializing/deserialing a string, coding/decoding a string, checking if a byte array contains valid UTF8 code, calculating the length of an encoded string.

`public Text()- constructor`

Class IntWritable

`public IntWritable(int value)- constructor`

Class StringTokenizer

The string tokenizer class allows an application to break a string into tokens. The tokenization method is much simpler than the one used by the `StreamTokenizer` class. The `StringTokenizer` methods do not distinguish among identifiers, numbers, and quoted strings, nor do they recognize and skip comments.

The set of delimiters (the characters that separate tokens) may be specified either at creation time or on a per-token basis.

An instance of `StringTokenizer` behaves in one of two ways, depending on whether it was created with the `returnDelims` flag having the value `true` or `false`:

- If the flag is `false`, delimiter characters serve to separate tokens. A token is a maximal sequence of consecutive characters that are not delimiters.
- If the flag is `true`, delimiter characters are themselves considered to be tokens. A token is thus either one delimiter character, or a maximal sequence of consecutive characters that are not delimiters.

A `StringTokenizer` object internally maintains a current position within the string to be tokenized. Some operations advance this current position past the characters processed.

A token is returned by taking a substring of the string that was used to create the `StringTokenizer` object.

The following is one example of the use of the tokenizer. The code:

```
StringTokenizer st = new StringTokenizer("this is a test");
while (st.hasMoreTokens()) {
    System.out.println(st.nextToken());
}
```

prints the following output:

```
this
is
a
test
```

set() method in Text Class

```
public void set(String string)
```

Set to contain the contents of a string.