

WebMapReduce Specifications (C++, RAB)

Revision 1.1

Class *Wmr*

WebMapReduce Library

Superclass: None

State Variables for Class Wmr:

Constructors for Class Wmr:

Methods for Class Wmr:

`emit`

Arguments:

`key` A **string** containing no TAB characters.

`value` String, either a **string** or a null-terminated `char *` value.

State Change: A key-value pair constructed from the arguments is appended to output

Return Value: None.

`strToLong`

Arguments:

`str` A **string** that represents an integer value that fits in a `long` variable.

Return Value: `int` value represented by that argument `str`.

`strToInt`

Arguments:

`str` A **string** that represents an integer value that fits in a `int` variable.

Return Value: `int` value represented by that argument `str`.

`strToDouble`

Arguments:

`str` A **string** that represents an `double` value.

Return Value: `double` value represented by that argument `str`.

`longToStr`

Arguments:

`num` A `long` value.

Return Value: `string` value that represents that argument `num`.

`intToStr`

4/21/10 (R
Brown): In
tial entry.

Arguments:

`num` A `int` value.

Return Value: `string` value that represents that argument `num`.

`doubleToStr`

Arguments:

`num` A `double` value.

Return Value: `string` value that represents that argument `num`.

`split`

Arguments:

`str` A `string` object to be split.

`delims` A `string` of delimiter characters. Default value is " " (space character).

State Change: A new `vector` of `string` is allocated and assigned strings of characters from `str` that are between characters that appear in `delims`. If multiple characters in `delims` appear consecutively, they are treated as a single delimiter.

Return Value: That newly allocated `vector`.

Class `WmrIterator`

Iterator class for WebMapReduce reducers.

An *iterator* provides a sequence of values, one at a time, in order. That sequence may be arbitrarily long, even too large to store in memory. This iterator class is designed for use in reducers, to provide values from the key-value pairs produced by mappers.

Note: This is an abbreviated spec, omitting constructors, destructor, and operators `=`, `==`, `!=`, `->`

Superclass: None.

State Variables for Class `WmrIterator`:

Constructors for Class `WmrIterator`:

Methods for Class `WmrIterator`:

`operator*`

Arguments: None.

State Change: None.

Return Value: A `string`, namely the current value for this iterator. An empty string is returned if all values for this iterator have already been traversed.

`operator++`

Arguments: None.

4/21/10 (R
Brown): In
tial entry.

State Change: If this instance of `WmrIterator` has any remaining values, this instance advances to the next value. If there are no remaining values, this instance of `WmrIterator` becomes equivalent to `WmrIterator::end`.

Return Value: This instance of `WmrIterator` in its state either *before* this call of `operator++` (for prefix operator call) or *after* this call of `operator++` (for postfix operator call).

Class Mapper

Class containing mapper method for a map-reduce computation

Superclass: None

State Variables for Class Mapper:

Constructors for Class Mapper:

Methods for Class Mapper:

`mapper`

Arguments:

`key` A `string` object containing no TAB characters.

`value` A `string` object.

State Change: 0 or more new key-value pairs are emitted for the “mapper” step of a map-reduce computation, using the method `Wmr.emit()`.

Return Value: None.

Class Reducer

Class containing reducer method for a map-reduce computation

Superclass: None

State Variables for Class Reducer:

Constructors for Class Reducer:

Methods for Class Reducer:

`reducer` Reducer method for a map-reduce computation

Arguments:

`key` A `string` object containing no TAB characters.

`value` A `WmrIterator` object consisting of strings.

State Change: 0 or more new key-value pairs are emitted for the “reducer” step of a map-reduce computation, using the method `Wmr.emit()`.

Return Value: None.

3/5/10 (R.
Brown): In
tial entry.

3/5/10 (R.
Brown): In
tial entry.