TextSorting

Generated by Doxygen 1.8.13

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Namespace Documentation

2.1 FullFileReader Namespace Reference

Change full file into variable and manupulate with it.

Functions

- void readFullFile (const char ∗FileName, char ∗∗text)
- size_t changeSlashesToNulles (char ∗text, char ∗∗∗indexes)
- void outputInFile (char ∗∗indexes, const char ∗FileName, size_t countOfLines, int typeOfWriting)

2.1.1 Detailed Description

Change full file into variable and manupulate with it.

2.1.2 Function Documentation

2.1.2.1 changeSlashesToNulles()

```
size_t FullFileReader::changeSlashesToNulles (
            char * text,
            char *** indexes )
```

change
to \0

Parameters

text	- variable pointing to text, where to be changed
indexes	- array where elemnts will index to the new tokens

Returns

count of elemnts in indexes

2.1.2.2 outputInFile()

```
void FullFileReader::outputInFile (
            char ** indexes,
            const char * FileName,
            size_t countOfLines,
            int typeOfWriting )
```

write text to file with order written in indexes array to the FileName file

Parameters

indexes	- array with order
text	- text where we will read
FileName	- file where to safe file
countOfLines	- count of elements in indexes
typeOfWriting	- parametres to open the file

2.1.2.3 readFullFile()

```
void FullFileReader::readFullFile (
            const char * FileName,
            char ** text )
```

read file to text pointer

Parameters

FileName	- name of the file to be read
text	- variable, that will point to text

2.2 StringSorter Namespace Reference

sort text

Functions

- bool isSpecial (char i)

- bool comparator (const char ∗a, const char ∗b, size_t firstStart, size_t secondStart, int firstIncrement, int secondIncrement, size_t firstEnd, size_t secondEnd)
- void freeText (char ∗text, char ∗∗indexes, size_t countOfLines)
- void sortMyFile (const char ∗SrcFileName, const char ∗DistFileName)

2.2.1 Detailed Description

sort text

2.2.2 Function Documentation

2.2.2.1 comparator()

```
bool StringSorter::comparator (
            const char * a,
            const char * b,
            size_t firstStart,
            size_t secondStart,
            int firstIncrement,
            int secondIncrement,
            size_t firstEnd,
            size_t secondEnd )
```

compare two objects

Parameters

a	- first StringToCompare object to be compared
b	- StringToCompare object to be compared compare to char∗
a	- first char∗
b	- second char∗
firstStart	- index to the first letter of first char∗
secondStart	- index to the first letter of second char∗
firstIncrement	- way to go through first char∗
secondIncrement	- way to go through second char∗
firstEnd	- index to the last letter of first char∗
secondEnd	- index to the last letter of second char∗

Returns

- true, if char∗ a starting from firstStart and iterating with + firstIncrement is bigger than b starting from secondStart iterating with + secondIncrement Empty char∗ -> at the end else, else

2.2.2.2 freeText()

```
void StringSorter::freeText (
            char * text,
            char ** indexes,
            size_t countOfLines )
```

free space where text is

Parameters

text	- pointer to first elemt of char
indexes	- pointer to array of indexes, where in left side \0 was put

2.2.2.3 isSpecial()

```
bool StringSorter::isSpecial (
            char i )
```

Parameters

i	

Returns

2.2.2.4 sortMyFile()

```
void StringSorter::sortMyFile (
            const char * SrcFileName,
            const char * DistFileName )
```

Text read to the variable,
changed to \0 and then outputs AscendingText outputs AscendingFromEndText outputs NormalText are called with appropriate parametres

Parameters

SrcFileName	- file where text must be got
DistFileName	- file where sorted text must be put

Index