

SimpleLogger

Wygenerowano przez Doxygen 1.8.13

Spis treści

1	README	1
2	Indeks klas	3
2.1	Lista klas	3
3	Dokumentacja klas	5
3.1	Dokumentacja szablonu klasy <code>buffer< T ></code>	5
3.1.1	Opis szczegółowy	5
3.1.2	Dokumentacja funkcji składowych	5
3.1.2.1	<code>addElem()</code>	5
3.1.2.2	<code>clearBuffer()</code>	6
3.1.2.3	<code>getFirst()</code>	6
3.1.2.4	<code>getLast()</code>	6
3.1.2.5	<code>isEmpty()</code>	7
3.2	Dokumentacja klasy <code>logMsg</code>	7
3.2.1	Opis szczegółowy	8
3.2.2	Dokumentacja konstruktora i destruktora	8
3.2.2.1	<code>logMsg()</code> [1/2]	8
3.2.2.2	<code>logMsg()</code> [2/2]	8
3.2.3	Dokumentacja funkcji składowych	8
3.2.3.1	<code>calcChecksum()</code>	8
3.2.3.2	<code>getMsg()</code>	9
3.2.3.3	<code>getPrefix()</code>	9
3.2.3.4	<code>getPriority()</code>	9
3.2.3.5	<code>printLog()</code>	9
3.2.3.6	<code>setMsg()</code>	9
3.2.3.7	<code>setPrefix()</code>	10
3.2.3.8	<code>setPriority()</code>	10
	Indeks	11

Rozdział 1

README

Simple logging library written for a programming class.

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

buffer< T >		
	Szablon klasy bufora	5
logMsg		
	Klasa wiadomości logowanych	7

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja szablonu klasy `buffer< T >`

Szablon klasy bufora.

Metody publiczne

- void `addElem` (T elem)
Metoda dodająca element na koniec bufora.
- void `clearBuffer` ()
Metoda czyszcząca bufor.
- T `getFirst` ()
Metoda zwracająca pierwszy element bufora.
- T `getLast` ()
Metoda zwracająca ostatni element bufora.
- bool `isEmpty` ()
Metoda sprawdzająca stan bufora.

3.1.1 Opis szczegółowy

```
template<class T>  
class buffer< T >
```

Szablon klasy bufora.

Szablon klasy bufora przechowującego logowane wiadomości.

3.1.2 Dokumentacja funkcji składowych

3.1.2.1 `addElem()`

```
template<class T >  
void buffer< T >::addElem (  
    T elem )
```

Metoda dodająca element na koniec bufora.

Metoda, która dodaje element na koniec bufora wiadomości

Parametry

<i>elem</i>	Element typu T, który ma być dodany do bufora
-------------	---

3.1.2.2 clearBuffer()

```
template<class T >
void buffer< T >::clearBuffer ( )
```

Metoda czyszcząca bufor.

Metoda, która usuwa wszystkie elementy bufora od pierwszego do ostatniego.

3.1.2.3 getFirst()

```
template<class T >
T buffer< T >::getFirst ( )
```

Metoda zwracająca pierwszy element bufora.

Metoda zwraca pierwszy element bufora, a także usuwa go z bufora.

Zwraca

Pierwszy element bufora.

3.1.2.4 getLast()

```
template<class T >
T buffer< T >::getLast ( )
```

Metoda zwracająca ostatni element bufora.

Metoda zwraca ostatni element bufora, a także usuwa go z bufora.

Zwraca

Ostatni element bufora.

3.1.2.5 isEmpty()

```
template<class T >
bool buffer< T >::isEmpty ( )
```

Metoda sprawdzająca stan bufora.

Metoda, która sprawdza czy bufor jest pusty czy też nie.

Zwraca

true gdy bufor jest pusty, false gdy nie

Dokumentacja dla tej klasy została wygenerowana z pliku:

- buffer.cpp

3.2 Dokumentacja klasy logMsg

Klasa wiadomości logowanych.

```
#include <logmsg.hpp>
```

Metody publiczne

- `logMsg` (std::string msg)
konstruktor tworzący obiekt wiadomość z ciągu znaków
- `logMsg` ()
Domyślny konstruktor.
- void `setPrefix` (std::string p)
Metoda dodająca prefiks.
- void `setPriority` (enum priority pr)
Metoda ustawiająca priorytet.
- void `setMsg` (std::string msg)
Metoda dodająca wiadomość
- int `getPriority` ()
Metoda zwracająca priorytet.
- std::string `getPrefix` ()
Metoda zwracająca prefiks.
- std::string `getMsg` ()
Metoda zwracająca wiadomość
- std::string `printLog` ()
Metoda zamieniająca wiadomość w ciąg znaków.
- void `calcChecksum` ()
Metoda obliczająca sumę kontrolną wiadomości.

3.2.1 Opis szczegółowy

Klasa wiadomości logowanych.

Klasa przechowuje pojedynczą wiadomość, która ma być logowana

3.2.2 Dokumentacja konstruktora i destruktora

3.2.2.1 logMsg() [1/2]

```
logMsg::logMsg (
    std::string msg )
```

konstruktor tworzący obiekt wiadomość z ciągu znaków

Konstruktor tworzy obiekt wiadomość z ciągu znaków stworzonego wcześniej przez metodą printLog. Sprawdza poprawność sumy kontrolnej. Jeśli się nie zgadza tworzona jest pusta wiadomość.

Parametry

<i>msg</i>	Ciąg znaków, na podstawie którego tworzony jest obiekt wiadomości.
------------	--

3.2.2.2 logMsg() [2/2]

```
logMsg::logMsg ( )
```

Domyślny konstruktor.

Domyślny konstruktor tworzący pustą wiadomość

3.2.3 Dokumentacja funkcji składowych

3.2.3.1 calcChecksum()

```
void logMsg::calcChecksum ( )
```

Metoda obliczająca sumę kontrolną wiadomości.

Metoda oblicza sumę kontrolną SHA1 wszystkich pól wiadomości, priorytetu, prefiksu i jej treści. Wartość suma jest przypisywana do pola checksum obiektu.

3.2.3.2 getMsg()

```
std::string logMsg::getMsg ( )
```

Metoda zwracająca wiadomość

Metoda zwraca treść logowanej wiadomości

Zwraca

ciąg znaków, treść wiadomości

3.2.3.3 getPrefix()

```
std::string logMsg::getPrefix ( )
```

Metoda zwracająca prefiks.

Metoda zwraca prefiks logowanej wiadomości

Zwraca

Ciąg znaków, prefiks wiadomości

3.2.3.4 getPriority()

```
int logMsg::getPriority ( )
```

Metoda zwracająca priorytet.

Metoda zwracająca wartość priorytetu logowanej wiadomości.

Zwraca

Jedna z trzech wartości priorytetu: error, warning, info

3.2.3.5 printLog()

```
std::string logMsg::printLog ( )
```

Metoda zamieniająca wiadomość w ciąg znaków.

Metoda zamienia cały obiekt w ciąg znaków możliwy do przesłania między klientem a serwerem

Zwraca

ciąg znaków, wiadomość w formacie możliwym do przesłania

3.2.3.6 setMsg()

```
void logMsg::setMsg (
    std::string msg )
```

Metoda dodająca wiadomość

Metoda pozwala ustawić treść logowanej wiadomości.

Parametry

<i>msg</i>	logowana wiadomość
------------	--------------------

3.2.3.7 setPrefix()

```
void logMsg::setPrefix (
    std::string p )
```

Metoda dodająca prefiks.

Metoda pozwala ustawić prefiks, który pojawi się przed logowaną wiadomością

Parametry

<i>p</i>	ciąg znaków, prefiks
----------	----------------------

3.2.3.8 setPriority()

```
void logMsg::setPriority (
    enum priority pr )
```

Metoda ustawiająca priorytet.

Metoda pozwala ustawić priorytet logowanej wiadomości na jeden z wybranych.

Parametry

<i>pr</i>	priorytet, wartości: error, warning, info
-----------	---

Dokumentacja dla tej klasy została wygenerowana z plików:

- logmsg.hpp
- logmsg.cpp

Skorowidz

- addElem
 - buffer, 5
- buffer
 - addElem, 5
 - clearBuffer, 6
 - getFirst, 6
 - getLast, 6
 - isEmpty, 6
- buffer< T >, 5
- calcChecksum
 - logMsg, 8
- clearBuffer
 - buffer, 6
- getFirst
 - buffer, 6
- getLast
 - buffer, 6
- getMsg
 - logMsg, 8
- getPrefix
 - logMsg, 9
- getPriority
 - logMsg, 9
- isEmpty
 - buffer, 6
- logMsg, 7
 - calcChecksum, 8
 - getMsg, 8
 - getPrefix, 9
 - getPriority, 9
 - logMsg, 8
 - printLog, 9
 - setMsg, 9
 - setPrefix, 10
 - setPriority, 10
- printLog
 - logMsg, 9
- setMsg
 - logMsg, 9
- setPrefix
 - logMsg, 10
- setPriority
 - logMsg, 10