SimpleLogger

Wygenerowano przez Doxygen 1.8.13

Spis treści

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:

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja szablonu klasy buffer< T >

Szablon klasy bufora.

`#include <buffer.hpp>`

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

elem	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. Jeśli bufor jest pusty zwraca pustą wiadomość.

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 plików:

- buffer.hpp
- 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.
- **operator bool** ()
- 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 pojedyńczą 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

msg	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

msg	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

p	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

pr	priorytet, wartości: error, warning, info

Dokumentacja dla tej klasy została wygenerowana z plików:

- logmsg.hpp
- logmsg.cpp

3.3 Dokumentacja klasy logServer

Klasa serwera logującego.

```
#include <logserver.hpp>
```

Metody publiczne

- void setRemotePort (int p)

  Metoda ustawiająca numer portu.
- void setLocalSocket (std::string filename)

  Metoda ustawiająca plik lokalnego gniazda.
- int **startLocallistener** ()
- int **startRemoteListener** ()
- int **stopLocalListener** ()
- int **stopRemoteListener** ()
- void setLogPath (std::string path)

  Metoda ustawiająca ścieżkę logów.
- bool enableBuffering (bool state)

  Metoda włączająca buforowanie.
- bool isBuffered ()

  Metoda zwracająca stan buforowania.
- int **saveLogs** ()

3.3.1 Opis szczegółowy

Klasa serwera logującego.

Klasa obsługująca serwer logujący przychodzące wiadomości.

3.3.2 Dokumentacja funkcji składowych

3.3.2.1 enableBuffering()

```
bool logServer::enableBuffering (
            bool state )
```

Metoda włączająca buforowanie.

Metoda pozwala włączać lub wyłączać buforowanie przychodzących wiadomości przed ich zapisem do logów.

Parametry

state	false - buforowanie wyłączone, true - buforowanie włączone

Zwraca

Aktualnie ustawiony stan buforowania

3.3.2.2 isBuffered()

```
bool logServer::isBuffered ( )
```

Metoda zwracająca stan buforowania.

Metoda zwraca informacje o tym czy włączone jest buforowanie przychodzących wiadomości do logowania.

Zwraca

true - buforowanie jest włączone, false - buforowanie jest wyłączone

3.3.2.3 setLocalSocket()

```
void logServer::setLocalSocket (
            std::string filename )
```

Metoda ustawiająca plik lokalnego gniazda.

Metoda ustawia ścieżkę do pliku, który ma być używany do obsługi lokalnych połączeń do serwera.

Parametry

filename	ciąg znaków, ścieżka do pliku

3.3.2.4 setLogPath()

```
void logServer::setLogPath (
            std::string path )
```

Metoda ustawiająca ścieżkę logów.

Metoda pozwala ustawić ścieżkę w której zapisywane będą logowane wiadomości.

Parametry

path	ścieżka, w której zapisywane mają być logi.

3.3.2.5 setRemotePort()

```
void logServer::setRemotePort (
            int p )
```

Metoda ustawiająca numer portu.

Metoda ustawia numer portu do nasłuchiwania zdalnych połączeń. Jeśli numer portu jest z poza zakresu 0,65536 ustawiana jest wartość domyślna 10000.

Parametry

p	numer portu, liczba całkowita

Dokumentacja dla tej klasy została wygenerowana z plików:

- logserver.hpp
- logserver.cpp

Skorowidz