LogTrace

Ein Beispiel zum Debuggen

Aufgabenstellung

- Erstellung einer Funktions-Bibliothek für diverse LOG und TRACE Ausgaben.
- Durch einfache Makros soll überall im Programm die Ausgabe:
  - von Fehler- und Informationsmeldungen (Text)
  - von Varaiblen (-inhalten)
  - in welcher Datei und Zeile
  - in welcher Funktion/Methode
- Jede dieser Ausgaben soll dabei mit einem Zahlenwert (1,2,...9) gewichtet werden können (LogLevel).
  - Damit soll bei Bedarf gesteuert werden, bis zu welchem LogLevel die Daten ausgegegben werden sollen.

Aufgabenstellung - 2

- Die Ausgabe erfolgt nach clog
- Das original Coding soll dabei möglichs wenig verändert werden. Hierzu soll die Funktionalität durch MAKROS zur Verfügung gestellt werden.
  - Über compiler Schalter soll sich die komplette Ausgabe abschalten lassen

Hilfsmittel Preprozessor

- Bedingtes Einfügen von Befehlszeilen abhängig von Existenz von Makros
  - beim Include von header
  - Ein/ausschalten von Makros

```
#ifndef TRACELOG_H
…
#endif
```

- Makros zur Erzeugung (Vorsicht) von coding.
  - Variablen können direkt X
  - Oder als Text #X (d.h. " x ")

```
#define MIN(X,Y)  (X<Y?X:Y)
#define OUT(T) cout << #T << T
```

- Vorsicht:
  - keine Operationen verwenden

- Vordefinierte Makros
  - __FILE__ Dateiname
  - __LINE__ Zeile

```
__FILE__
__LINE__
```

Vorgehensweise

- Die benötigten Funktionen sollen alle eine möglichst "breite" Schnittstelle haben
  - D.h. alle benötigten Parameter annehmen (evtl. mit Default Werten)
  - Evtl. kann die Funktion auch überladen sein (um andere Typen zu erlauben)
- Damit das "Handling" nicht zu kompliziert wird soll die Funktion hinter "einfachen" Makros verborgen werden
  - Beispiel:
  - #define LOGTRACE (1,TEXT) LogTrace (1, __FILE__ , __LINE__ , TEXT);
  - Damit kann dann sehr einfach das Coding im NICHT-Verwendungsfall gesäubert werden.
  - #define LOGTRACE (1,TEXT)

Funktionssammlung

- Neben einer Textausgabe, sollen auch andere Funktionen zur Ausgabe von Variablen verwendet werden.
  - LogTrace (int Level, char * File, int Number , char * Text)
  - LogTrace (int Level, char * File, int Number , char * Text , long Var)
  - .....
  - LogTraceSetLevel

Beispiel LogTrace

Logtrace.h

```
#ifdef LOGTRACE_H
#define LOGTRACE(L,TEXT) LogTrace (L , __FILE__ , __LINE__, TEXT );
#define LOGTRACE_LV(L,X) LogTrace (L , __FILE__ , __LINE__, #X , X);
#else
#define LOGTRACE(L,TEXT)
#define LOGTRACE_LV(L,X)
#endif
```

Logtrace.cpp

```
void LogTrace (int level, char * File , int Line ,char *text)
{
   if (level < LOGLEVEL)
        clog << level << " - " << text << "  -  ["
             << File << ":" << Line << "]" << endl;

}
```