

## **Thema, Ziele:** Code Bloat bei Templates

### **Aufgabe 1:** Vermeiden von Code Bloat bei Templates, Code Hoisting

In dieser Aufgabe soll untersucht werden, wie mittels Code Hoisting der bei Templates häufig entstehende Code Bloat vermieden werden kann.

Die folgenden Optionen der GNU-Compiler könnten nützlich sein:

- E Precompile only, der Output wird auf stdout geschrieben
- S Assembleroutput, ohne Objectfile erzeugen
- c nur compilieren
- O0 keine Optimierung
- O1 Optimierungsstufe 1 (siehe g++ - Help für Details)
- O2 Optimierungsstufe 2
- O3 Optimierungsstufe 3
- Os Optimierung auf Codegrösse

- a) Untersuchen Sie den Code im Verzeichnis ./Vorgabe/StackTemplate. Verschaffen Sie sich einen Überblick und werden Sie vertraut mit der Templateprogrammierung.
- b) Compilieren Sie den Code mit Optimierungsstufe 0 und 3. Verwenden Sie dabei sowohl den aktuellen Compiler (Version 4), als auch Version 3 (g++-3.exe). Untersuchen Sie jeweils den entstandenen Assemblercode sowie die Codegrösse. Achten Sie vor allem darauf, ob die Funktionen aufgerufen werden (mit call) oder inline sind. Sind die Funktionen einfach oder mehrfach vorhanden?
- c) Ändern Sie den Code nun so ab, dass im Hauptprogramm ein zweiter int-Stack mit unterschiedlicher Grösse definiert wird.  
Compilieren Sie den Code wiederum unter Verwendung beider Compilerversionen mit Optimierungsstufe 0 und 3. Untersuchen Sie jeweils den entstandenen Assemblercode sowie die Codegrösse. Achten Sie darauf, ob die Funktionen aufgerufen werden (mit call) oder inline sind. Sind die Funktionen einfach oder mehrfach vorhanden?
- d) Für diese Teilaufgabe die Version 3 des Compilers verwenden (g++-3.exe).  
Code Hoisting: Sie haben bestimmt festgestellt, dass bei der Compilerversion 3 die Funktionen nun doppelt vorhanden sind. Die Klasse StackUI lassen wir unverändert. Analysieren Sie, welche Teile in der Klasse Stack unabhängig von der Grösse sind. Lagern Sie alle diese Teile in eine Basisklasse aus.  
Compilieren Sie den Code wiederum mit Optimierungsstufe 0 und 3. Untersuchen Sie jeweils den entstandenen Assemblercode sowie die Codegrösse. Achten Sie darauf, ob die Funktionen aufgerufen werden (mit call) oder inline sind. Sind die Funktionen einfach oder mehrfach vorhanden?
- e) Versuchen Sie dasselbe mit der Klasse StackUI. Setzen Sie auch hier die Version 3 ein.