Übungen zur Vorlesung

"Objektorientiertes Programmieren im Wissenschaftlichen Rechnen"

Dr. Olaf Ippisch, Ole Klein

Abgabe am 17. 06. 2014 in der Vorlesung

ÜBUNG 1 STATISCHER VS. DYNAMISCHER POLYMORPHISMUS

Wir wollen Laufzeitunterschiede von Programmen bei Verwendung von statischem bzw. dynamischen Polymorphismus untersuchen. Als Beispielprogramm betrachten wir wieder die numerische Integration.

Statt der Verwendung von dynamischen Polymorphismus können wir die Quadraturregel und/oder die zu integrierende Funktion auch mittels statischem Polymorphismus (Templates) auswählen.

1. Schreiben Sie drei zusätzliche Varianten ihres bereits entwickelten Integrationsprogrammes mit folgenden Eigenschaften:
   (a) Statische Auswahl der Quadraturregel, dynamische Auswahl der Funktion.
   (b) Dynamische Auswahl der Quadraturregel, statische Auswahl der Funktion.
   (c) Statische Auswahl der Quadraturregel, statische Auswahl der Funktion.
2. Messen sie in den vier Hauptprogrammen die für die Integration benötigte Zeit. Am einfachsten geht dies, indem Sie sich den Header `timer.hh` von der Homepage herunterladen. Die dort enthaltene Klasse `Timer` besitzt zwei Funktionen. `Timer::elapsed` gibt die benutzte CPU-Zeit in Sekunden an und `Timer::reset` setzt den Zähler für die Zeit auf Null. Sie können sich auch in den Namespace `std::chrono` einarbeiten, der deutlich mehr Funktionaliät hat, dieser einfache Timer reicht für uns jedoch aus.
3. Messen Sie die für die Integration von $\int_{-3}^{13} 2t^2 + 5$ mit 10002 Stützstellen benötigte Zeit für alle vier Versionen einmal ohne Optimierung (Compiler-Option "-O0") und einmal mit Optimierung (Compiler-Option "-O3"). Messen Sie sowohl für die Simpson- als auch die Trapezregel, und geben Sie die Ergebnisse in einer Tabelle an!
4. Gibt es Laufzeitunterschiede? Falls ja, interpretieren und begründen Sie diese! Lohnt es sich in dieser Situation, sich Gedanken um die zu wählende Form von Polymorphismus zu machen? Falls ja, mit welchen Argumenten?

20 Punkte

Wie immer gilt: Kommentieren Sie Ihr Programm. Erklären Sie was Sie tuen.