

Thema, Ziele: Listen, Algorithmen und Komplexitätstheorie

Aufgabe 1: Klasse für die Speicherung von Messwerten

Die Messwertliste **ist** eine Liste, deshalb ist aus objektorientierter Sicht eine Vererbungsbeziehung vorzuziehen.

Das Eclipse-Projekt ist unter `./Loesung/MeasureListInh` zu finden.

Die Variante mit einer Aggregation finden Sie unter `./Loesung/MeasureListAgg`

Aufgabe 2: Komplexitätsbetrachtungen

Bestimmen Sie den benötigten Aufwand der folgenden Algorithmen in O-Notation. Alle Algorithmen berechnen die Potenz $c=a^b$

In der Funktion `potenz1()` wird die while-Schleife b mal durchlaufen, d.h. dieser Algorithmus ist $O(b)$.

Die Funktion `potenz2()` wird b mal rekursiv aufgerufen, weitere Schleifen sind nicht vorhanden, d.h. dieser Algorithmus ist $O(b)$.

In der Funktion `potenz3()` wird b bei jedem Schleifendurchlauf halbiert. Die Schleife wird ungefähr $\lg(b)$ mal durchlaufen (\lg = logarithmus dualis, Zweierlogarithmus). `potenz3()` ist demnach $O(\lg b)$.

Aufgabe 3: Implementation eines dynamischen Stacks mit Hilfe einer verketteten Liste (optional)

Der Stack ist **keine** Liste, er benutzt nur eine für die Speicherung der Daten. Eine Vererbung kommt deshalb nicht in Frage, Aggregation ist die richtige Variante. Da der Stack beliebige Daten speichern können soll, muss er unbedingt als Templateklasse implementiert werden.

Das Eclipse-Projekt ist unter `./Loesung/DynaStack` zu finden.