



# Programming by Contract

WS 2003/2004 – Übungsblatt 12 (optional)

Ausgabe: 2. Februar 2004

## Aufgabe 1. *Contracts in Kindklassen, Forts.*

Das Beispiel

[http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PgmByContr0304/name\\_list2.cc](http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PgmByContr0304/name_list2.cc)

demonstriert den Einsatz von Nana bei der Vererbung (wie werden Vor- und Nachbedingungen sowie Invarianten der Elternklasse in der Kindklasse berücksichtigt?).

Listen Sie alle erwähnenswerten Implementierungshinweise für diese Anwendung tabellarisch in Form von „Hilfestellungen“ (getrennt jeweils für Elternklasse und Kindklasse) auf und erläutern Sie sie in eigenen Worten.

## Aufgabe 2. *Quicksort-Frame-Regeln*

Stellen Sie die in der Vorlesung besprochenen alternativen Möglichkeiten der Frame-Regel-Spezifikation der Funktion `quicksort()` gegenüber und diskutieren Sie jeweils Vor- sowie Nachteile.

## Aufgabe 3. *Konzeption einer Klasse mit Contracts*

Konzipieren Sie eine (C++-)Klasse `SIMPLE_QUEUE` mit den Attributen/Methoden `Konstruktor(en)`, `basic queries`, ... im Funktionsumfang `number_of_items`, `capacity`, `is_empty`, `is_full`, `first`, `put`, `remove`, `invariant` und spezifizieren Sie vollständig (auch die Frame-Regeln).

## Aufgabe 4. *Copy-Konstruktor*

Ergänzen Sie `mydictionary` aus `dictionary.cc` um einen Copy-Konstruktor. Spezifizieren Sie diesen „vollständig“ und implementieren Sie. Planen Sie auskommentierte Testanweisungen, die die Verletzung der Nachbedingungen mindestens je einmal auslösen.