



Programming by Contract

SS 2007 – Übungsblatt 11

Ausgabe: 5. Juli 2007

Abgabe: bis spätestens 12. Juli 07
in der Vorlesung

Aufgabe 1. *nana-sfg*

Benutzen Sie den modifizierten **nana short form generator**, um aus der Beispielquelle

dictionary4.cc

den Contract herauszuextrahieren und benutzen Sie diesen sowie den Original-C++-Quellcode zur Bearbeitung der folgenden Aufgaben.

Aufgabe 2. *Framebedingung dictionary*

Warum sind die Nachbedingungen des Kopierkonstruktors in **dictionary4.cc** nur bedingt für einen (Kunden-)Contract geeignet? Reformulieren Sie einen (evtl. noch nicht ausführbaren) Contract nur mit Hilfe der grundlegenden Observatoren.

Kritisieren Sie analog die Berechnung von **old_keys** in der Methode **put()**.

Wozu sind die virtuellen set-Methoden **+** und **-** eingeführt worden?

Versuchen Sie die Framebedingungen der Methode **put()** sowie der Methode **remove()** mit Hilfe einer Kopie von ***this** ausführbar zu machen.

Aufgabe 3. *Längenbeschränkter Container*

Ändern Sie den Contract aus Aufgabe 1 in einen solchen für ein a-priori in der Länge beschränktes Dictionary um. (Es gibt keinen Default-Konstruktor, sondern nur einen Konstruktor mit einem Parameter, der die maximale Länge spezifiziert.)

Implementieren Sie die neue Klasse und testen Sie!

Aufgabe 4. *Methode keys()*

Ergänzen Sie die Klasse `mydictionary` (aus `mydictionary4.cc`) um einen neuen grundlegenden Observator

```
set<KEY> keys()
```

der die STL-Menge der Schlüssel des Exemplars der Klasse `mydictionary` als Funktionsergebnis liefert.

Redesignen Sie die gesamte Klasse unter Benutzung dieses grundlegenden Observators. Welche ehemals grundlegenden Observatoren werden nun zu abgeleiteten Observatoren? Wie ändert sich deren Contract?

Aufgabe 5. *quicksort_unique*

Spezifizieren Sie eine Funktion

```
int quicksort_unique(double v[], int n);  
// Nach Funktionsaufruf enthält v[]  
// in den Komponenten 0..(Wert des Funktionsergebnisses)  
// alle Werte aus dem ursprünglichen v[] in aufsteigender  
// Reihenfolge, jedoch keinen doppelt
```

durch Angabe von Vor- und Nachbedingungen in nana-Schreibweise.