



**BERGISCHE  
UNIVERSITÄT  
WUPPERTAL**

Prof. Dr. Hans-Jürgen Buhl  
Praktische Informatik/Numerik

Fachbereich C  
Mathematik und Naturwissenschaften,  
Mathematik und Informatik

E-MAIL buhl@math.uni-wuppertal.de

WWW www.math.uni-wuppertal.de/~buhl

DATUM 19. Mai 2015

## **Softwarequalität**

**SS 2015 – Übungsblatt 6**

**Ausgabe: 20. Mai 2015**

**Abgabe bis 3. Juni 2015 an: <mailto:125319@uni-wuppertal.de>**

### **Aufgabe 1. C++17?**

Auf der Seite 12 von

<http://www.diku.dk/forskning/performance-engineering/Generic-programming/Slides/concepts.pdf>

wird das Requirement `LessThanComparable` mit den Axiomen `Transitivity`, `Antisymmetry`, ... spezifiziert.

Erklären Sie in eigenen Worten, welche Eigenschaften hier semantisch gefordert werden. Vergleichen Sie mit <http://www.sgi.com/tech/stl/LessThanComparable.html> und Seite 431 von <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2011/n3242.pdf>.

### **Aufgabe 2. `name_list`**

Erläutern Sie in eigenen Worten die zweite und dritte Nachbedingung der Methode `name_list::put()` (Materialsammlung Seite 17).

Erweitern Sie die Vorbedingung auf `true` und ergänzen Sie zwei weitere Nachbedingungen der Form `ENSURE(not_in_list || ...)`.

### **Aufgabe 3. `Queue`**

Erklären Sie umgangssprachlich Zeile für Zeile die denotationale Spezifikation 1.3.2 der Materialsammlung (Spezifikation eines `Queue`).

### **Aufgabe 4. `Kindklassen`**

Können in C++ Kindklassen einzelne Attribute oder Methoden der Elternklasse nicht besitzen? Wenn ja: Wie kann das realisiert werden, und ist es sinnvoll? (Unterscheiden Sie C++03 und C++11)

### **Aufgabe 5. C++14/17**

Was ist gemäß

<http://en.wikipedia.org/wiki/C%2B%2B14>

unter C++14 zu verstehen? Wann ist es erschienen? Was ändert sich an der „Function return type deduction“? Welche Vorteile hat die automatische Typ-Inferenz gemäß [http://en.wikipedia.org/wiki/C++11#Type\\_inference](http://en.wikipedia.org/wiki/C++11#Type_inference) bei der Benutzung von Iteratoren?

Was ist gemäß

<http://en.m.wikipedia.org/wiki/C%2B%2B17>

in C++17 mindestens zu erwarten?