



**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 4. Juni 2014

Formale Methoden

SS 2014 – Übungsblatt 8

Ausgabe: 4. Juni 2014

Abgabe bis 18. Juni 2014 an: dsavvidi+fm@studs.math.uni-wuppertal.de

Aufgabe 1. *Java Vector Obsolete/Optional Data in Java8 and C++11*

Lesen Sie die Artikel

<http://stackoverflow.com/questions/1386275/why-is-java-vector-class-considered-obsolete-or-deprecated>,

<http://stackoverflow.com/questions/5763838/java-6-se-are-vectors-obsolete>,

<http://stackoverflow.com/questions/1792134/a-colleague-said-dont-use-java-util-vector-anymore-why-not>

und

<http://forums.netbeans.org/post-78501.html>.

Warum ist also in welcher Arbeitsumgebung in Java die Collection-Klasse `Vector` ähnlich Abbildung 2.2 der Materialsammlung nicht mehr für die Implementierung von UML [*]-Assoziationen empfehlenswert?

Welche Klassen sollten statt dessen verwendet werden?

Die Vielfachheit [0..1] optionaler Datenfelder oder Assoziationsenden wird heute durch

<http://docs.oracle.com/javase/8/docs/api/java/util/Optional.html> (Java8)

beziehungsweise

<http://stackoverflow.com/questions/16860960/how-should-one-use-stdoptional> (C++11)

modelliert. Vergleichen Sie beide Lösungsansätze miteinander. Wie wurde es früher häufig gemacht und zu welchen Problemen führte es mitunter?

Aufgabe 2. *Tutorial UML Associations in Java7*

Erstellen Sie aus

http://www.pm.inf.ethz.ch/education/seminars/verification/archive/2004_2005/seminar/slides/6.12.Mapping_UML.pdf

und

<http://www.torsten-horn.de/techdocs/java-collections.htm#TraditionelleUndNeueCollections>

sowie den Java-Teilen von

http://www.jot.fm/issues/issue_2003_09/article4.pdf

eine kurze Tabelle, die die empfehlenswerten Java7-Collections zur Implementierung von "multivalued association ends" der Eigenschaften {ordered}, {ordered, nonunique}, {unordered}, {unordered, nonunique} mit oder ohne qualifizierten Zugriff beinhaltet. Ergänzen Sie eine Zeile für optionale [0..1]-Rollenenden.

Aufgabe 3. *Java 8 Collections and UML Associations*

Java8 hat unter anderem den Methoden-Satz der verfügbaren Container-Klassen erweitert:

<http://www.oracle.com/technetwork/java/javase/8-whats-new-2157071.html>.

Beschreiben Sie die neuen Fähigkeiten. Modifizieren Sie, falls nötig, Ihre Tabelle aus Aufgabe 2 für Java8.

Aufgabe 4. *Eclipse Papyrus/CDT C++ Code Generator*

Lesen und referieren Sie die in

http://wiki.eclipse.org/Codegen_description

beschriebenen C++-Codegeneration-Eigenschaften von CDT. Welche der in

<http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Integration-und-Migration-von-IT-Systemen/Software-Reengineering>

beschriebenen Reengineering-Techniken kann man mit ihnen nutzen?

Aufgabe 5. *Sparx Codegenerator/UML Class Diagram Editor C++ Code Generator*

Wie kann man in

http://www.sparxsystems.com/enterprise_architect_user_guide/9.2/software_engineering/generatesourcecode.html

mittels der „Source Code Options“ und der „C++ Options“ beziehungsweise der „Java Options“ den vom Sparx-Codegenerator erzeugten Quellcode insbesondere in Bezug auf "multivalued association ends" beeinflussen?

Wie im "[UML Class Diagram Editor C++ Generator](#)"? Was versteht man hier insbesondere unter G-Templates? Wie ist der generierte Implementierungscode für UML-Assoziationen und andere UML-Objekte beeinflussbar?