Mock exam 2017

Prof. Dr. Ralf Lämmel
University of Koblenz-Landau
Faculty of Computer Science
Software Languages Team
Logistics of actual exam
20 July 2017, regular lecture slot

• Mandatory registration until 18 July 2017

• 12.00-12.15: Take seats, check IDs, etc.

• 12.15-13.15: Exam
  • Closed book exam
  • Tasks in English
  • Answers in German or English

• Resit end of Winter Semester
Topics

1. Design patterns
2. Data programming
3. Metaprogramming
4. Mining Software Data
5. Distributed systems
6. Documentation of technologies

Exam topics
= lecture topics
= lab topics
= assignment topics

1 task per topic.
3 points per task.
18 points in total.

Grading scale*:
9 points 4.0
...
18 points 1.0

* With adjustments to account for problematic tasks.
Types of tasks

• Code comprehension (3/6 tasks):
  • “Explain the following code.”

• Concept comprehension (3/6 tasks):
  • “Explain the concept XYZ.”

All tasks solely relate to lecture and lab material available online. [http://www.softlang.org/course:ptt17](http://www.softlang.org/course:ptt17)

Concept comprehension tasks rely on lecture material.

Code comprehension tasks are mostly drawn from lab material.

The space for answers will be limited. You need to be concise.
1st Topic
“Design patterns”

• Concept comprehension:
  • Summarize the problem addressed by the pattern X.
  • Sketch a scenario in which to use the pattern.
  • Provide a class diagram for the pattern’s solution.
  • What are the roles of the classes and methods in the pattern’s solution.

• Relevant patterns:

Abstract Factory, Object Adapter, Class Adapter, Decorator, Observer,
Proxy, Singleton, Template Method, Visitor
Sample task “Design patterns”

Consider the solution of the Visitor pattern. What’s the problem addressed by the pattern? What is the role of the classes and methods shown? How many visitConcreteElement-like methods do we need in an application of the pattern?
2nd Topic
“Data programming”

• Code comprehension.

• Relevant technologies:
  • org.jdom (DOM, JDOM)
  • javax.json (JSON)
  • java.sql (JDBC)
  • JAXB
  • JPA
Sample task “Data programming”

```java
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "artikel", propOrder = {"name", "preis", "lagerbestand"})
public class Artikel {
    @XmlElement(required = true)
    protected String name;
    protected double preis;
    @XmlElement(required = true)
    protected BigInteger lagerbestand;
    @XmlAttribute(name = "identifier", required = true)
    @XmlJavaTypeAdapter(CollapsedStringAdapter.class)
    @XmlID
    @XmlSchemaType(name = "ID")
    protected String identifier;
...
}
```

The code was generated by JAXB. What is the overall purpose of the annotations? Explain the semantics of two distinct annotations.
3rd Topic
“Metaprogramming”

• Concept comprehension:

• Relevant concepts
  • Fluent APIs
  • Parsing with ANTLR
  • Java reflection

Template processing is omitted from exam.
The code constructs a finite state machine. Which techniques for fluent APIs are used? Briefly explain each technique based on the sample code. (3 techniques needed.)
4th Topic
“Mining Software Data”

• Code comprehension.

• Relevant techniques/technologies:
  • Java Parser
  • Regular expression matching
  • Nltk (examples from lab)
The code is part of an analyzer for Java files. What is accomplished by the code? (No detail explanation!) What would typically follow the statement at hand? What sort of compilation is referred to? Why is it useful?
5th Topic
“Distributed systems”

• Code comprehension.

• Relevant technologies:
  • JMS, Flask

RMI and Java WebServices are omitted from exam.
Sample task “Distributed systems”

api.add_resource(ArtikelList, '/artikel')
api.add_resource(Artikel, '/artikel/<artikel_id>')
6th Topic
“Documentation of technologies”

• Concept comprehension:
  • Identify and explain some aspect in a given model.

• Relevant models:
  • Models from lecture
Consider the following model for XML data binding (a la JAXB). What are the roles of the language entities in the model? (BTW, the textual MegaL notation may be used as well.)