Exam Winter Semester 2007/2008

February 27\textsuperscript{th}, 2008

Institute for Software Systems, TUHH

Name: _______________________________________________________________

Student Number: _________________________

Rules:
1. The exam is closed book. That means that the only things you are allowed to have on your desk or use during the exam are pens and the exam itself.
2. All phones off. A switched on phone is considered cheating.
3. Keep your eyes on your own work.
4. Cheating will cause you to fail this exam.

Additional information:
- The exam will take 90 minutes.
- Each problem is associated with a number of points. This is also the amount of time in minutes we expect you to take for the answer (the total sum is 75).
- Please put your student identification as well as a passport/official id card on the table. We need to check these.
- Don’t forget to put your name and student number on each page.
- If you draw into existing diagrams, please think \textit{before} you draw to keep your answer readable.
- We took the utmost care to make the English and German version semantically identical. In case of doubt, you may inspect both versions on the front desk.
- We have more paper, should you need some, ask.

\begin{center}
\textbf{Do not start reading this exam until instructed to do so}
\end{center}
Task 1: Software Development

In the context of model-driven software development, explain the following terms:
(a) Platform Independent Model (PIM) (3 pts)
(b) PIM to PSM conversion (3 pts)
(c) Give an example of a PIM to PSM conversion. (2 pts)

Task 2: Project Management

(a) Explain the main ideas of PERT activity networks. (5 pts)
Task 3: Algorithms

The class model depicted in Figure 1 is used as part of a larger application to track trains. One of the properties that all trains in this application should fulfill is shown as an OCL invariant. However, there are other constraints that should also be fulfilled, and operations that should be supported. Given their English formulation below, provide for each item below a Java implementation.

(a) The wagons linked over the association next should not establish a cycle. (5 pts)

(b) Assume that an additional boolean field (isFirstClass) is available for each wagon, indicating whether the wagon belongs to the 1st or 2nd class. Specify that no 2nd class wagon may be followed by a 1st class wagon. (5 pts)

(c) A function receives as arguments two trains A and B (i.e., two linked lists of wagons). Write the function body to return a new train C, obtained by picking as first wagon the first wagon of A, as second wagon the first of B, as third wagon the second of A, as fourth wagon the second of B, and so on. (5 pts)

(d) Write a function that receives a train and returns another containing the same wagons but in reverse order. (5 pts)

```
context Wagon
inv lastWagonHasLightsOn : f()

context Wagon::f() : Boolean
def : if next.oclIsUndefined() then hasLightsOn else next.f() endif
```

Figure 1
Task 4: Collections.

The diagram on Figure 2 depicts a fragment of a class hierarchy for collections.

(a) Draw (in the correct place) the interface Set. (3 pts)

(b) Why is the operation `elementAt(int):Object` available for objects of type `List` but not for `Collection` objects? (3 pts)

(c) In which type should the operation `iterator()` be declared? (This operation returns an iterator object). (3 pts)

(d) The declaration of the operation `elementAt(int)` in class `ArrayList`, is an example of overriding, overloading, or interface implementation? Justify your answer. (3 pts)
Task 5: Java programming language

Please indicate whether the following statements are true or false. (Points are discounted for wrong answers, so if in doubt you may consider leaving fields blank instead of randomly ticking a checkbox) (12 pts)

<table>
<thead>
<tr>
<th>Statement</th>
<th>true</th>
<th>false</th>
</tr>
</thead>
<tbody>
<tr>
<td>An array of Strings allocates a fixed number of characters for each array element.</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>The Java Virtual Machine can only execute programs whose source code was written in Java.</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Every class can be subclassed.</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Every class has a superclass. An exception is java.lang.Object</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>If a class has two or more constructors then they must be overloaded</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Different classes can implement the same interface.</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Java has a limited form of reflection, by which existing declarations can be queried but no declarations can be added at runtime</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>All user-defined classes are subtypes of java.lang.Object</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>If a class is empty (i.e., no attributes or methods are declared) then the compiler generates no bytecodes for it.</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>A String is not an object type but a primitive type</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>The addition of Generics in Java 5 made the language weakly-typed</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Java cannot perform bit-level manipulation because it’s not a low-level programming language.</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

Task 6: Alloy

Describe:

(a) the main aspects of the Alloy language, (5 pts)
(b) the kind of analysis that it makes possible, and (3 pts)
(c) the advantages of such analyses with respect to testing. (2 pts)
Task 7: Frameworks

(a) Define *Design Pattern* and provide three examples of design patterns you know, listing for each the problem it addresses. (5 pts)

(b) Are there frameworks for testing? If so, mention one or more examples. If not, justify why such idea does not make sense. (3 pts)