Classroom exercise 1

7 May 2004

The classroom exercise intends to help you self-evaluate your knowledge and skills and let us gain knowledge about the performance of our students. The setup resembles the situation you will encounter during the final exam. The assistants will be happy to clarify any problems with the formulation of the tasks, but will not solve the tasks for you. This exercise will be corrected and graded by your assistant; the grade will not have any influence on the final exam.

Duration: 2 hours (without break)

Please solve this exercise alone. No material allowed. Don’t forget to write your name and first name on top of each page.

1. Objects (20 points)

Feature categories
- Give two classifications of features. You may want to use a picture. (8 points)
- Give the two O-O principles corresponding to these two classifications. (4 points)

Garbage collection
- What does it mean for a Garbage Collector (GC) to be consistent? (2 points)
- What does it mean for a GC to be complete? (2 points)
- Explain the pros and cons of automatic garbage collection. (4 points)

2. Genericity (6 points)

Class vs. Type
- Give the definition of a type. (2 points)
- Is there a one to one correspondence between a type and a class? (2 points)
- Explain the difference between a class and a type. (2 points)
3. Design by Contract (60 points)

**Different kinds of contracts**
- List the different kinds of assertions, and explain the role of each. (12 points)

**Benefits**
- Describe four key benefits of Design by Contract. Explain. (8 points)

**Class correctness**
- What does it mean for a class to be correct? (2 points)
- Give a formal definition. (2 points)

**Contract violation**
- What does a precondition violation express? (2 points)
- What does a postcondition violation express? (2 points)
- What does a class invariant violation express? (2 points)

**Feature calls in assertions**
Consider a function `my_function` such as

```
my_function is  -- Do something.
  require
    some_query
do
  -- Something here
  ensure
    another_query
end
```

Explain the rules governing `some_query` and `another_query` to be able to use them in the routine contracts. (4 points)

**Introducing contracts**
Add contracts to the following class `ACCOUNT` and complete the comments if necessary. (16 points)

```plaintext
class
    ACCOUNT
create
    make
```
feature \texttt{\{NONE\}} – Initialization

\texttt{make (an\_amount: \textbf{like} balance)} \texttt{is}
\begin{verbatim}
  -- Set balance to an\_amount.
  do
    balance := an\_amount
  end
\end{verbatim}

feature -- Access

\begin{verbatim}
balance: \texttt{INTEGER}
  -- Account balance

Minimum\_balance: \texttt{INTEGER} \texttt{is} 1000
  -- Minimum amount of money on the account
\end{verbatim}

feature -- Deposit

\begin{verbatim}
deposit (an\_amount: \textbf{like} balance) \texttt{is}
  -- Add an\_amount to current balance.
  do
    balance := balance + an\_amount
  end
\end{verbatim}

feature -- Withdrawal

\begin{verbatim}
withdraw (an\_amount: \textbf{like} balance) \texttt{is}
  -- Subtract an\_amount from current balance.
  do
    balance := balance - an\_amount
  end
\end{verbatim}

feature -- Status report

\begin{verbatim}
may\_withdraw (an\_amount: \textbf{like} balance): \texttt{BOOLEAN} \texttt{is}
  -- May an\_amount be withdrawn from the account?
  do
    Result := (balance-an\_amount \geq minimum\_balance)
  end
\end{verbatim}

\textbf{Contract extraction}

Here is an extract of the documentation provided with the .NET Framework for method Insert of class System.Collections.ArrayList:
public virtual void Insert (int index, Object value);

Inserts an element into the ArrayList at the specified index.

Parameters
index
The zero-based index at which value should be inserted.
value
The Object to insert.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentOutOfRangeException</td>
<td>index is less than zero.</td>
</tr>
<tr>
<td></td>
<td>- or - index is greater than Count.</td>
</tr>
<tr>
<td>NotSupportedException</td>
<td>The ArrayList is read-only.</td>
</tr>
<tr>
<td></td>
<td>-or- The ArrayList has a fixed size.</td>
</tr>
</tbody>
</table>

Remarks

If Count already equals Capacity, the capacity of the list is doubled by automatically reallocating the internal array before the new element is inserted.

Write a complete specification of this feature (in Eiffel syntax) including preconditions and postconditions. (The preconditions and postconditions may involve calls to other class functions; if so, spell out the functions.) (10 points)

4. Inheritance (14 points)

Terminology

Define the following terms:
- Parent (2 points)
- Child (2 points)
- Heir (1 point)
- Ancestor (2 points)
- Proper ancestor (1 point)
- Descendant (2 points)
- Proper descendant (1 point)
- Instance (2 points)
- Direct instance (1 point)