Exercise 4: Inheritance

Hand-out: 7 May 2004
Due: 14 May 2004

Please solve this exercise alone.

Summary: Inheritance

Principle: Describe a new class not from scratch but as extension or specialization of one existing class — or several in the case of multiple inheritance (which you will see in the next lecture).

- **Redefinition**: A class may change an inherited feature.
- **Polymorphism**: An entity may have different forms at run-time.
- **Dynamic binding**: Effect of a feature call \( x.f \) depends on run-time form of \( x \).

Here is a typical example:

```plaintext
p: POLYGON
r: RECTANGLE     -- Assume RECTANGLE inherits from POLYGON.
t: TRIANGLE       -- Assume TRIANGLE inherits from POLYGON.
x: INTEGER

... if some_test then
    p := r
else
    p := t
end

x := p.perimeter       -- Assume class POLYGON has a feature perimeter
                        -- returning an INTEGER.

-- If some_test is true, p.perimeter uses the version of perimeter from RECTANGLE.
-- If some_test is false, p.perimeter uses the version of perimeter from TRIANGLE.
```
1. The stacks of Hanoï

(This exercise in an extract of *Object-Oriented Software Construction, 2nd edition*, by Bertrand Meyer; page 869.)

Assume a deferred class \textit{STACK} with a procedure \textit{put} to push an element onto the top, with a precondition involving the boolean-valued function \textit{full}.

Now consider the famous problem of the Towers of Hanoï, where disks are stacked on piles – the towers – with the rule that a disk may only be put on a larger disk.

To do

Is it appropriate to define the class \textit{HANOI\_STACK}, representing such piles, as an heir to \textit{STACK}? If so, how should the class be written? If not, can \textit{HANOI\_STACK} still make use of \textit{STACK}? Write the class in full for the various possible solutions; discuss the pros and cons of each, state which one you prefer, and explain the rationale for your choice.

Hint

The boolean query \textit{full} could also be called \textit{extendible}; as you study the exercise you will note that the choice of name may affect the appeal of various possible solutions.

To hand in

Hand in the text of class \textit{HANOI\_STACK} and explanations.

2. Where do the iterators belong?

(This exercise in an extract of *Object-Oriented Software Construction, 2nd edition*, by Bertrand Meyer; page 870.)

Would it be a good idea to have iterator features (\textit{while\_do} and the like) included in classes describing the data structures on which they iterate, such as \textit{LIST}? Consider the following points:

- The ease of applying iterations to arbitrary \textit{action} and \textit{test} routines, chosen by the application.
- Extendibility: the possibility of adding new iteration schemes to the library.
- More generally, respect of object-oriented principles, in particular the idea that operations do not exist by themselves but only in the relation to certain data abstractions.

To hand in

Hand in the (detailed) answer to the above question.
3. About dialog

To do
Write a class \textit{ABOUT\_DIALOG}. The dialog layout should be as follows:

The dialog title should be “My fancy about dialog”. You can choose whatever image you want. The message should be:

\begin{itemize}
\item My fancy about dialog
\item Copyright (C) 2004 <your name here>
\item All rights reserved
\item ETH Zentrum
\item CH-8092 Zurich
\item Switzerland
\item Electronic mail: <your email address here>
\end{itemize}

Hint
Make your class \textit{ABOUT\_DIALOG} a descendant of the class \textit{EV\_DIALOG} from EiffelVision2.

To hand in
Hand in the text of class \textit{ABOUT\_DIALOG}. 