Exercise 2

24.10.2001 **Due date: 07.11.2002**

Exercise 1

Currency converter: This exercise uses the example of a very simple currency converter to demonstrate the application of a basic framework construction principle ? Unification Principle

The currency converter program takes one input representing an amount of money expressed in Euro and produces one output representing the same amount of money expressed in US Dollar. The conversion factor from Euro to US Dollar is assumed to be equal to 0.98.

The currency converter program is parameterized by the rounding policy used to round the US Dollar amount.

Rounding policy: The following rounding policies are foreseen:

- **RoundingPolicyDefault**: rounds the results to four decimal digits (i.e., 34.97655 to 34.9766)
- **RoundingPolicy10**: rounds the result to one decimal digit (i.e., 34.97655 to 35.0)
- **RoundingPolicy100**: rounds the result to two decimal digits (i.e., 34.97655 to 34.98)

The currency converter program should be designed to allow easy switching from one rounding policy to another.

Compile-time adaptation: The rounding policy is decided statically (at compile time) and therefore the Unification design principle is used to ensure flexibility in the choice of the rounding policy. The currency converter program should include a simple user interface consisting of the following elements:

- an input field where the user enters the amount of money in EUR to be converted to USD.
- an output field where the result of the conversion is displayed
- a button to do the conversion

Exercise 2

This exercise introduces modifications to the version of the *currency converter program* built in exercise 1.

Run-time adaptation: The currency converter program should be designed to allow easy switching from one rounding policy to another. In particular, two versions of the currency converter program will be produced:

- The rounding policy is decided dynamically (at run time) and therefore the Separation design

principle is used to ensure flexibility in the choice of the rounding policy

- Use an interface, which represent the different rounding policies. The rounding policies implement the interface and the rounding method.

The currency converter program should include a simple user interface consisting of the following elements:

- an input field where the user enters the amount of money in EUR to be converted to USD
- an output field where the result of the conversion is displayed
- a button to do the conversion
- a choice menu where the rounding policy is selected

Exercise 3

This exercise introduces a modifications to the version of the *currency converter program* built in exercise 2.

In this modification, users will be given the option of dynamically loading a new rounding policy.

The program as built in exercise 2 allowed users to select at run-time one out of three predefined rounding policies. The new version of the program will let user specify a rounding policy by name and the corresponding class will then be dynamically loaded and linked into the program. It will be assumed that the name of the class to be loaded is not known at the time the program is developed.

This version of the program will use the meta-information services of the C# language.

Implementation of the second modification of the currency converter program will require an extension of the user interface of the currency converter program to include an input field where users specify the class implementing the desired rounding policy as a string.
