

Quick Reference Sheet SEMI-OOS

School Adequate Development Environment for Modelling and Implementation of object-oriented Software

© 2006-2011 Robert Pütterich

A. General Issues

First, make sure that on your computer a matching Java compiler with an interpreter (package "JDK", at least version 5) is installed. You can download it free of charge from the internet.

The program SEMI-OOS has no installation program as it consists only of a single file, the Java archive "System_SEMIOOS.jar". The file can be executed similar to the execution of an "exe file".

When you first start the program, this quick reference and the license agreement are being extracted from the Java archive.

SEMI-OOS has no main-window menu. All the basic functions are available by the command buttons package in the upper area of the screen, which is hereafter referred to as control console. At the bottom the respective work area is being displayed. Context menus can be displayed with the right mouse button.

After having first started the program, please click on the "Preferences / Einstellungen" button of the control console.

SEMI-OOS requires the location of the Java compiler on your system. You can use the automatic search function or you can manually chose the file "javac.exe", which usually is to be found in the installation directory... \ Programs \ Java \ jdk..... \ bin.

In the lower area you can chose the language, the system font and the user level. In the beginner mode no class attributes and class methods, no array lists and no access permissions can be displayed.

The settings are stored in the file "semioos.pref".

B. Documentation of the work areas and GUIs

Selection by the command buttons on the control console:

a) "New Project", "Open Project", "Save Project" and "Save Project As"

A project consists of only a single file with the extension ".semi", which is managed in the usual way with these command buttons.

b) "Class"

You use this work area for the definition of classes with attributes, methods and related parameters.

The text box is used to label the class, stored content can be found in the box under the command button trio "new", "save" and "delete".

Storage is hierarchical, i.e. if you save a class, all the according attributes and methods, stored in each field selection, are assigned to this class. The same is true when you save a method for the corresponding parameters.

The GUI for the definition of the algorithms for the individual methods can be reached via the command button "method" of the control console.

c) "GUI - Class"

For each new window SEMI-OOS creates a class of its own; in doing so corresponding attributes for the objects which have been drawn are being created. In addition, for each command button a method is defined, which is being executed when you press the command button.

With the class editor (command button "Class", see above) attributes and methods can be added to a GUI - Class.

d) "Method"

The algorithms of the methods are determined in accordance with the Nassi-Shneiderman Structograms, a way of modelling algorithms (not dependant on programming languages).

SEMI-OOS offers a minimal but effective selection of algorithmic structures, which are selected on the left (orange marker) and are inserted or added via a right mouse click and the menu "insert element". If you click on an element of the structogram with the left mouse button, the editor for this element is opened in the lower half of the window frame.

When editing the instructions or conditions of structogram elements, text blocks are being offered. The blocks also include a lot of very useful basic methods of a SEMI-OOS integrated class "BM".

In the editor section you get "upwards" with the cursor key to the selection field; The "CR" key adds the selected building block, and with the "Esc" key you can always choose to leave the field.

When you start typing in a building block, the selection box assumes the task of completing the text, which then can be completed with the "CR" key.

If only single-figure building blocks are being inserted, e.g. when entering a variable name, and the building block is unambiguous, it will automatically be adopted after the character has been typed in.

If there is a correct and complete expression together, the command button "Apply" gets active and can be pressed. If "Apply" is active, it will be pressed automatically, if you continue with the example. If "Apply" is inactive and you continue with the example, changes will be dropped.

e) "Inheritance"

This work area is used to define inheritance relationships between classes.

GUI - Classes can not bequeath under SEMI-OOS, but they inherit from a pre-defined class "Frame". This class is the equivalent of Java's "JFrame".

f) "Start Command"

Supported by text blocks the program's entry-level command is determined.

g) "Implementation"

The command buttons on the left offer the following options:

"SEMIOOS Implementation"

- Implementation of the project in a newly developed language, which is as intuitively understandable as possible

"Java Implementation"

- Implementation of the project in the Java language (version 5.0 and higher)

"Compile Java Implementation"

- creation of a Java source code file, which bears the name of the project file preceded by "System" and followed by the suffix ".java"
- translation of the source code into byte code ("class" files)

"Generate and Execute Java archive (*.JAR)"

- compression of the "class" files to an executable Java archive ("jar" file) and subsequent execution
- Cleaning up of the "class" files

"Java Implementation as Applet"

- Implementation of the project in Java as an applet. To do so, exactly one GUI - Class has to be defined and the Start Command has to begin with "generateNewObject_NameOfGUIClass ("

"Compile Java-Applet Implementation"

- creation of a Java source code file with the applet implementation, which bears the name of the GUI - Class with the extension „.java“
- translation of the source code into byte code ("class" files)

"Generate Java-Applet-Archive (*.JAR) and HTML-Framework"

- compression of the "class" files to a Java applet archive ("jar" file)
- Generation of an HTML file which can be opened externally with a browser, and which also bears the name of the GUI - Class with the extension ".html" and carries a scaffold for the integration of applets

h) "Independent Structogram"

For each method you can always generate a independent structogram which can be defined according to your own preferences. This structogram is then only illustrative and of no significance for the implementation.

i) "Preferences / Einstellungen"

For details, see *General Issues*.