

# **Kurzreferenz SEMI-OOS**

Schulgemäße Entwicklungsumgebung zur Modellierung und Implementierung objektorientierter Software

© 2006-2011 Robert Pütterich

## **A. Allgemeines**

Stellen Sie zunächst sicher, dass auf Ihrem Computer ein passender Java-Compiler mit Interpreter (Paket „JDK“, mindestens Version 5) installiert ist, den Sie kostenlos aus dem Internet beziehen können.

Das Programm SEMI-OOS hat kein Installationsprogramm, da es nur aus einer einzigen Datei, dem Java-Archiv „System\_SEMIOOS.jar“, besteht. Diese Datei ist ähnlich wie eine „exe-Datei“ ausführbar.

Beim ersten Start wird aus dem Java-Archiv diese Kurzreferenz und der Lizenzvertrag extrahiert.

SEMI-OOS besitzt kein Hauptfenster-Menü. Alle Grundfunktionen sind über das Befehlsschaltflächen-Paket im oberen Bereich, welches im Folgenden Steuerungskonsole genannt wird, erreichbar. Im unteren Bereich wird die jeweilige Arbeitsoberfläche angezeigt. Kontextmenüs erreichen Sie über die rechte Maustaste.

Klicken Sie bitte nach dem ersten Start zunächst auf die Befehlsschaltfläche „Einstellungen / Preferences“ der Steuerungskonsole.

SEMI-OOS benötigt den Standort des Java-Compilers auf Ihrem System. Sie können diesen automatisch suchen lassen oder manuell die Datei „javac.exe“ wählen, die sich in der Regel im Installationsverzeichnis ... \ Programme \ Java \ jdk..... \ bin befindet.

Im unteren Bereich kann neben der Sprache und der System-Schriftart gewählt werden, ob die Oberfläche von SEMI-OOS für Einsteiger oder Fortgeschrittene dargestellt werden soll. Im Einsteigermodus können keine Klassenattribute und –methoden, keine Feldlisten und keine Zugriffsberechtigungen festgelegt werden.

Die Einstellungen werden in der Datei „semioos.pref“ fixiert.

## **B. Dokumentation der einzelnen Arbeitsbereiche**

**Auswahl über die Befehlsschaltflächen der Steuerungskonsole:**

**a) „Projekt neu“, „Projekt öffnen“, „Projekt speichern“ und „Projekt speichern unter“**

Ein Projekt besteht aus nur einer einzigen Datei mit der Endung „.semi“, welche im üblichen Rahmen mit den genannten Befehlsschaltflächen verwaltet wird.

## **b) „Klasse“**

Sie verwenden diesen Arbeitsbereich zur Definition von Klassen mit Attributen, Methoden und zugehörigen Parametern.

Im Textfeld wird jeweils die Bezeichnung eingegeben, gespeicherte Inhalte finden Sie im Auswahlfeld unter dem Befehlsschaltflächentrio „neu“, „speichern“ und „löschen“.

Die Speicherung erfolgt hierarchisch, d.h. falls Sie eine Klasse speichern, werden alle zugehörigen Attribute und Methoden, die im jeweiligen Auswahlfeld abgelegt sind, dieser Klasse zugeordnet. Gleiches gilt beim Speichern einer Methode für die zugehörigen Parameter.

Die Oberfläche zur Definition der Algorithmen zu den Methoden erreichen Sie über die Befehlsschaltfläche „Methode“ der Steuerungskonsole.

## **c) „Oberflächen-Klasse“**

Für jedes erstellte Fenster legt SEMI-OOS eine Klasse an, wobei für die im Fenster gezeichneten Objekte entsprechende Attribute erzeugt werden.

Zudem wird für jede Befehlsschaltfläche eine Methode, die beim Drücken der Befehlsschaltfläche ausgeführt wird, definiert.

Mit dem Klasseneditor (Befehlsschaltfläche „Klasse“, s.o.) können weitere Attribute und Methoden zu einer Oberflächen-Klasse hinzugefügt werden.

## **d) „Methode“**

Die Festlegung der Algorithmen der Methoden erfolgt mit dem Struktogramm nach Nassi-Shneiderman, einer (programmiersprachenunabhängigen) Modellierungstechnik für Algorithmen.

Bei SEMI-OOS wird eine minimale aber effiziente Auswahl an algorithmischen Strukturen angeboten, welche auf der linken Seite angewählt (orange Markierung) und per rechtem Mausklick und dem Menüpunkt „Struktogrammelement einfügen“ ein- bzw. angefügt werden. Klickt man mit der linken Maustaste auf ein Element des Struktogramms, so öffnet sich für dieses Element der Editor im unteren Fensterbereich.

Beim Editieren der Anweisungen bzw. Bedingungen von Struktogrammelementen werden Textbausteine angeboten. Die Bausteine beinhalten auch eine Menge sehr nützlicher Basismethoden einer in SEMI-OOS integrierten Klasse BM.

Im Editor gelangt man mit der Cursor-Taste „nach-oben“ in das Auswahlfeld, die Eingabetaste fügt den markierten Baustein ein und mit der Esc-Taste kann das Auswahlfeld stets verlassen werden.

Wird mit dem Eintippen eines Bausteines begonnen, so erfüllt das Auswahlfeld die Aufgabe der Textvervollständigung, welche mit der Eingabetaste dann abgeschlossen werden kann. Werden nur einstellige Bausteine eingegeben, z.B. bei der Eingabe eines Variablennamens und ist der Baustein eindeutig, so wird automatisch nach dem Tippen des Zeichens der Baustein übernommen.

Wurde ein korrekter und vollständiger Ausdruck zusammengesetzt, so wird die Befehlsschaltfläche „übernehmen“ aktiv, welche dann gedrückt werden kann. Ist „übernehmen“ aktiv, so wird sie beim Fortfahren z.B. mit der Bearbeitung eines anderen Struktogrammelementes automatisch gedrückt. Ist „übernehmen“ nicht aktiv, so werden aktuelle Änderungen beim Fortfahren verworfen.

## **e) „Vererbungsbeziehung“**

Diese Arbeitsfläche dient zur Definition von Vererbungsbeziehungen zwischen Klassen. Oberflächenklassen können unter SEMI-OOS nicht vererben, sie erben aber von einer vorgegebenen Klasse „Hauptfenster“. In Java entspricht diese der Klasse „JFrame“.

## **f) „Startanweisung“**

Unterstützt durch die Vorgabe von Textbausteinen wird hier die Einstiegsanweisung in das Programm festgelegt.

## **g) „Implementierung“**

Die Befehlsschaltflächen auf der linken Seite bieten folgende Optionen an:

### **„SEMIOOS Implementierung“**

- Implementierung des Projektes in einer neu entwickelten, möglichst intuitiv verständlichen Sprache

### **„Java Implementierung“**

- Implementierung des Projektes in der Sprache Java (ab Version 5.0)

### **„Java Implementierung compilieren“**

- Erzeugung einer Java-Quellcode-Datei, die den Namen der Projektdatei mit vorangestelltem „System\_“ und der Endung „.java“ trägt
- Übersetzung des Quellcodes in Bytecode („class-Dateien“)

### **„Java Archiv (\*.JAR) erzeugen und ausführen“**

- Zusammenfassung und Komprimierung der „class-Dateien“ zu einem ausführbaren Java-Archiv („jar-Datei“) und anschließende Ausführung
- Aufräumen der „class-Dateien“

### **„Java Implementierung als Applet“**

- Implementierung des Projektes in Java als Applet. Dazu muss genau eine Oberflächenklasse definiert sein und die Startanweisung mit „erzeugeNeuesObjekt\_NameDerOberflächenklasse(“ beginnen.

### **„Java-Implementierung als Applet compilieren“**

- Erzeugung einer Java-Quellcode-Datei mit der Applet-Implementierung, die den Namen der Oberflächenklasse mit der Endung „.java“ trägt
- Übersetzung des Quellcodes in Bytecode („class-Dateien“)

### **„Java-Applet-Archiv (\*.JAR) und HTML-Gerüst erzeugen“**

- Zusammenfassung und Komprimierung der „class-Dateien“ zu einem Java-Applet-Archiv („jar-Datei“)
- Erzeugung einer extern mit einem Browser zu öffnenden HTML-Datei, die ebenfalls den Namen der Oberflächenklasse mit der Endung „.html“ trägt und ein Gerüst zur Einbindung des Applets enthält

## **h) „unabhängiges Struktogramm“**

Sie können zu jeder Methode neben dem Struktogramm, welches in die Implementierung übernommen wird, ein frei formulierbares Struktogramm erstellen. Dieses Struktogramm dient nur der Veranschaulichung und hat für die Implementierung keine Bedeutung.

## **i) „Einstellungen / Preferences“**

Erläuterungen dazu finden Sie unter „Allgemeines“.