

# Programmablauf - eine Übersicht

Im folgenden soll der Programmablauf von TeSeT als eine Art Übersicht dargestellt werden. In der Darstellung befinden sich mehrere Hinweise auf den Quellcode und daher sind einige Notationserklärungen notwendig. TeSeT wurde in Smalltalk programmiert, einer Programmiersprache, die eine im Vergleich zu gängigeren Sprachen eine ungewöhnliche Syntax aufweist:

Klasse>>methode weist auf 'methode' die sich in 'Klasse' befindet.

Methoden, die ein Argument nehmen, enden auf einem Doppelpunkt:

methodenName: Argument

Nimmt die Methode mehrere Argumente, wird der Name wie gefolgt aufgesplittet:

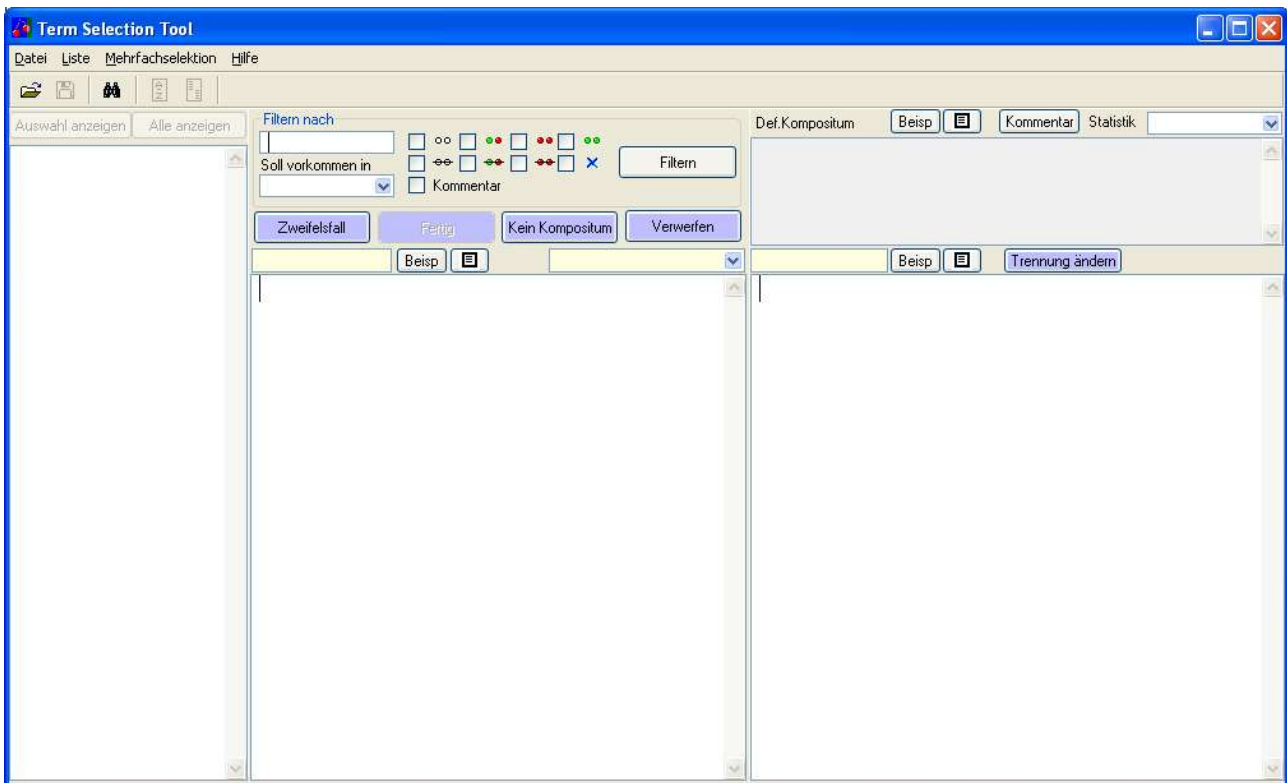
methodenName: argument1 fortsetzungMethodenName: argument2

Für die genauere Beschreibung der Klassen sehen Sie sich bitte die Programmdokumentation an: [teset\\_progrDoku.html](#)

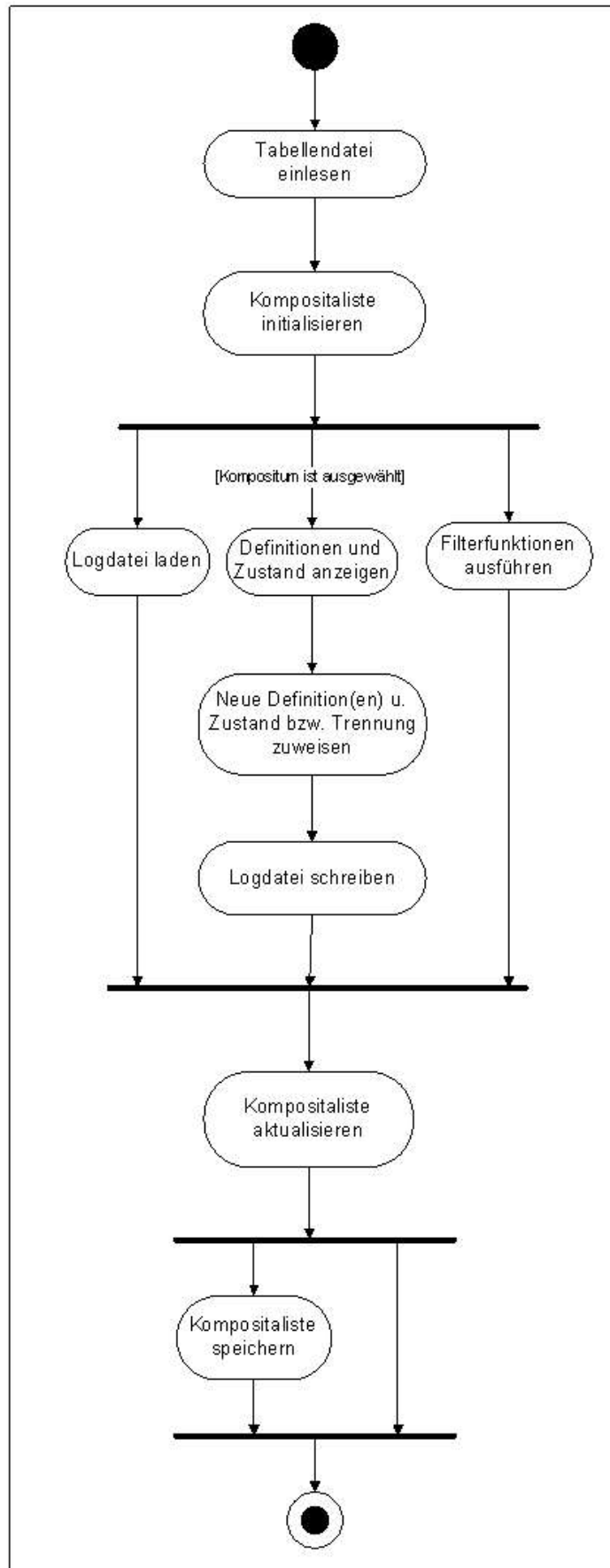
## 1. Starten des Programms - Öffnen des Applikationsfensters

Die grafische Benutzeroberfläche wurde mit Hilfe von *UIPainter* erstellt, einem Paket der VisualWorks Entwicklungsumgebung, das Klassen zur Erstellung einer GUI zur Verfügung stellt. Die GUI wurde quasi per drag-and-drop gezeichnet und die zugehörige Applikationsklasse von der abstrakten Smalltalk-Klasse *ApplicationModel* abgeleitet. Diese Klasse stellt ein Gerüst für Applikationsklassen zur Verfügung, die *UIBuilder* für die Generierung der GUI benutzen. Somit müssen abgeleitete Klassen lediglich um anwendungsspezifische Methoden ergänzt werden.

Nach dem Starten des Programms erscheint das Applikationsfenster:



## 2. Mögliche Abläufe - Aktivitätsdiagramm



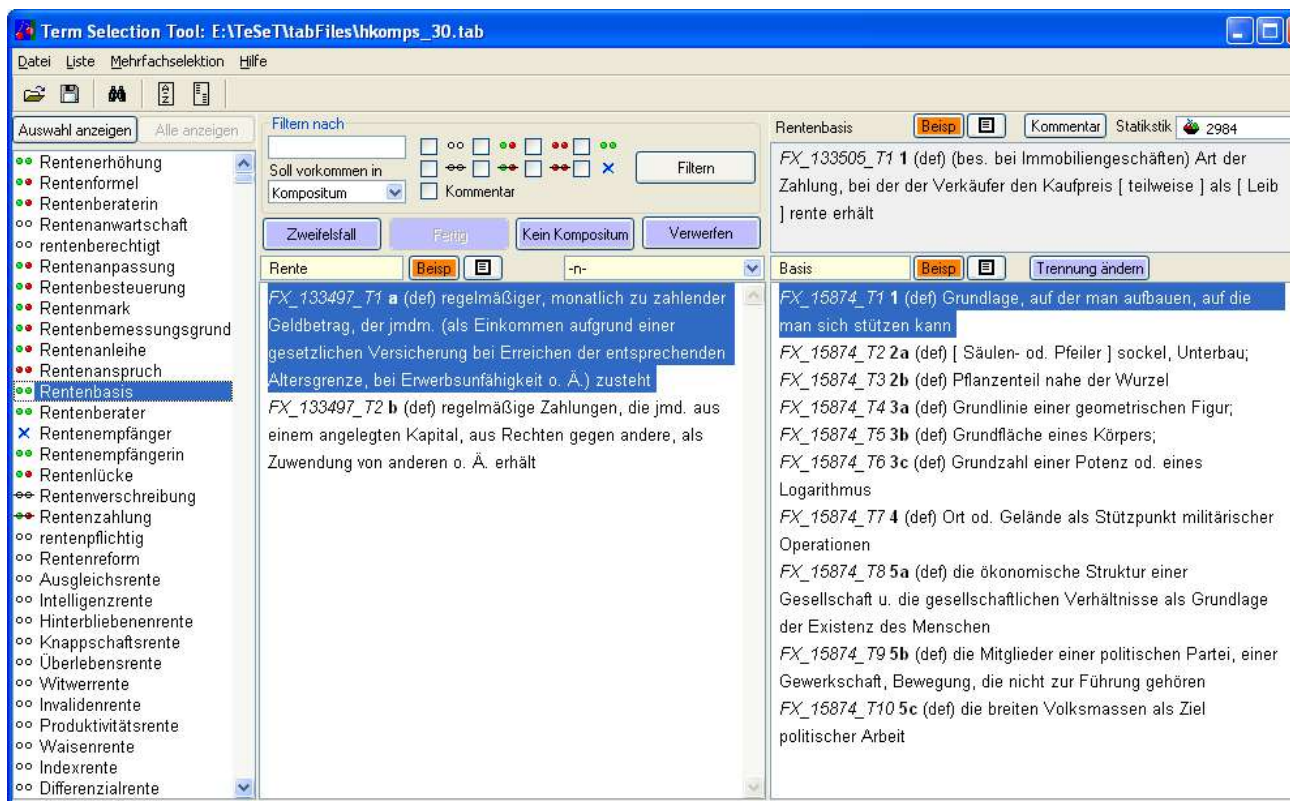
Als erstes muss eine Datei eingelesen werden, die eine Kompositaliste beinhaltet. Diese Dateien sind im Tabellenformat und beinhalten verschiedene Informationen zu den Komposita. Eine Datei, die mit Hilfe von TeSeT bearbeitet werden soll, muss ein bestimmtes Format haben und in einer Headline spezifiziert sein.

Ist die eingelesene Datei gültig, wird die TeSeT interne Kompositaliste (TSTool>>compounds) mit dem Inhalt der Datei initialisiert. Diese Liste ist eine Instanz der Klasse TSCollection, die wiederum aus Instanzen der Klasse TSCompound besteht. Jede Instanz der Klasse TSCompound repräsentiert ein Kompositum mit allen notwendigen Informationen. Die beiden Hauptkomponenten eines Kompositums werden als Instanzen der Klasse TSCompoundElement abgebildet.

Der Benutzer hat nun drei Möglichkeiten, um weiterzugehen:

- 1) Eine Logdatei laden. Wurden aus irgendeinem Grund die Ergebnisse des letzten Arbeitens mit TeSeT nicht gespeichert, können sie durch das Laden der Logdatei wieder hergestellt werden.
- 2) Filterfunktionen ausführen. Jede Kompositaliste enthält typischerweise ca. 3000 Komposita. Um diese Liste übersichtlicher zu machen, stehen verschiedene Filterfunktionen zur Verfügung.
- 3) Die Termzuordnung der Komposita vornehmen. Diese verläuft folgendermaßen:

Wenn ein Kompositum angeklickt wird, erscheinen Definitionstexte in den Fenstern für das Bestimmungswort, für das Basiswort, sowie für das Kompositum selbst. Falls das Kompositum schon bearbeitet worden ist, so ist es als solches einerseits in der Liste gekennzeichnet und andererseits werden die früher ausgewählten Definitionen blau markiert. Es gibt verschiedene Möglichkeiten ein Kompositum zu bearbeiten, diese sind einzeln in der Benutzeranleitung beschrieben.



Nach jedem der vorangegangenen drei Schritte wird die Kompositaliste aktualisiert.

Wenn der Benutzer sich für das Speichern seiner Arbeit entscheidet, wird die Kompositaliste mit allen zugehörigen Informationen in eine Tabelle geschrieben. Der Benutzer kann dabei wählen, ob er die aktuelle Datei überschreibt oder ob er eine neue Datei erzeugt.