

## Anwendungen

**Lehre:** EToys (Squeak), SqueakBot, BotsInc, Scratch ...

**Multimedia:** Sophie, OpenCroquet, Plopp ...

**Web Entwicklung:** Seaside, Aida, Komanche, Swazoo ...

**Persistenz:** Objektorientierte Datenbanken (Magma, GemStone), Relationale Datenbanken (MySQL, PostgreSQL), Objektrelationales Mapping (Glorp).



Eine Mal-Session mit Plopp

## Glossar

**Image:** Die Smalltalk-Umgebung beinhaltet einen persistenten Objektspeicher, das Image. Im Image werden alle Klassen- und Methoden-Objekte des Systems und somit sein gesamter Source-Code gespeichert, aber auch der aktuelle Zustand aller Werkzeug-Objekte wie Klassenbrowser, Inspektoren oder Debugger.

**Virtuelle Maschine:** Eine virtuelle Maschine ist ein Programm, das fähig ist, andere Programme auszuführen und damit die Portabilität von Anwendungen erleichtert.

**Reflektion:** Eine Sprache nennt man reflektiv, wenn sie Mechanismen beinhaltet, mit deren Hilfe man Code zur Laufzeit betrachten und ändern kann.

**Dynamische Typisierung:** Einige Sprachen zwingen die Entwickler den Typ jeder Variable (Integer, String ...) anzugeben. Dies nennt man statische Typisierung. Mit dynamischer Typisierung limitiert man Variablen nicht auf einen gegebenen Typ.

## Bücher

- Eine Vielzahl von frei verfügbaren Büchern:  
<http://stephane.ducasse.free.fr/FreeBooks.html>
- Smalltalk im Allgemeinen
  - *Smalltalk with Style* (Edward Klimas, Suzanne Skublics and David A. Thomas, frei verfügbar)
  - *Smalltalk by Example: the Developer's Guide* – (Alec Sharp, frei verfügbar)
- Squeak im Besonderen
  - *Squeak by Example* – (2007, frei verfügbar)
  - *Powerful Ideas in the Classroom* (BJ Allen-Conn und Kim Rose)

## Veranstaltungen

- Konferenzen der European Smalltalk User Group (ESUG). Seit 1993, industrielle und akademische Smalltalker treffen sich in einem europäischen Land.  
<http://www.esug.org/conferences>
- Jährliche Konferenz in Nord-Amerika, organisiert vom STIC (<http://www.stic.st>)  
<http://www.smalltalksolutions.com/>

## Internet

- Offizielle Squeak-Website:  
<http://www.squeak.org>
- EToys:  
<http://www.squeakland.org>
- Seaside:  
<http://www.seaside.st>
- Wiki:  
<http://wiki.squeak.org>
- News:  
<http://news.squeak.org>
- Deutschsprachig:  
<http://squeak.de>

# Smalltalk

eine rein  
objektorientierte  
Programmiersprache  
und dynamische  
Entwicklungsumgebung



Wichtige Konzepte von Smaltalk

Squeak ist ein Smaltalk-Dialekt. Smaltalk ist eine *objektorientierte* Sprache, *dynamisch typisiert*, mit einer minimalen Syntax, die in *fünfzehn Minuten* gelernt werden kann. Ein wesentlicher Vorteil der Smaltalk-Sprache ist ihre *Kohärenz*:

- Alles ist ein Objekt: Klassen, Methoden, Zahlen, etc.
- Es existieren nur eine kleine Anzahl von Syntaxregeln und keine Ausnahmen.

Smaltalk läuft auf einer *virtuellen Maschine*. Die Entwicklung erfolgt in einem *Image*, in dem alle Objekte "leben" und modifiziert werden können.

Smaltalk-Syntax

Reservierte Wörter	
n11	Undefiniertes Objekt
true und false	Boolesche Objekte
self	Aktuelles Objekt
super	Aktuelles Objekt im Kontext der Superklasse
thisContext	Laufzeit-Stack der aktuellen Methode
Reservierte Zeichen	
Zuweisung = (oder ⇐) ~ (oder ↑)   var1 var2 var3   Deklaration dreier temporärer Variablen	
\$a	Zeichen a
#(abc 123)	Array mit zwei Literalen #abc und 123
· (punkt)	Stoppzeichen eines Ausdrucks
;	Nachrichten-Kaskadierung
[ ]	Code-Block (auch ein Objekt)
"Kommentar"	'Zeichenkette'

Der Aufruf einer Methode erfolgt durch das Versenden einer Nachricht an ein Empfängerobjekt, welches anschließend wiederum ein Objekt zurückliefert. Es existieren nur unäre, binäre und Schlüsselwort-Nachrichten.

**Unäre Nachrichten.** Eine unäre Nachricht hat kein Argument.

```
array := Array new.  
array size.
```

Das erste Beispiel erzeugt eine neue Instanz von Array, indem es die unäre Nachricht new an die Klasse Array sendet. Das zweite Beispiel liefert die Größe dieses leeren Arrays, also 0.

**Binäre Nachrichten.** Eine binäre Nachricht hat nur ein Argument, wird durch ein Symbol bezeichnet und oft für arithmetische Ausdrücke verwendet.

```
3 + 4.  
'Hello', ' World'.
```

Die Nachricht + wird an das Objekt 3 mit 4 als Parameter geschickt. Im zweiten Fall wird die Nachricht , an den String 'Hello' mit ' World' als Parameter geschickt.

**Schlüsselwort-Nachrichten.** Im Unterschied zu Sprachen wie C, C++ , Java oder Python werden Parameter nicht mittels Klammern übergeben, sondern in Anlehnung an die natürliche Sprache mit in den Methodenaufruf eingebaut. Die Parameter werden dazu zwischen sogenannte Schlüsselwörter positioniert, jeweils durch einen Doppelpunkt getrennt.

```
'Smaltalk' a11ButFirst: 5.  
3 to: 10 by: 2.
```

Das erste Beispiel ruft die Methode a11ButFirst: mit dem Parameter 5 auf einem String auf. Das zweite Beispiel ruft die Methode to:by: mit den Parametern 10 und 2 zwischen den Schlüsselwörtern to: und by: auf der Integerzahl 3 auf.

Blöcke

Blöcke sind Code in eckigen Klammern, der nicht sofort ausgeführt wird, sondern bedingt, wiederholt oder später.

```
#('Hallo ' Welt')  
do: [:wort | Transcript show: wort].
```

Dieses Beispiel schickt die Nachricht do: an ein Feld von Zeichenketten und übergibt dabei einen Block. Die Methode

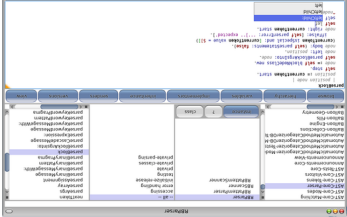
do: führt den Block mit jedem Feldelement aus, das als Parameter word übergeben wird und nacheinander 'Hallo ' und 'Welt' ausgibt.

Entwicklungsumgebung

Die meisten Smaltalk-Implementierungen behalten eine Entwicklungsumgebung, die es erlaubt, im Source-Code zu navigieren und mit Objekten zu interagieren. Diese Umgebungen behalten viele Werkzeuge, die alle in Smaltalk selbst implementiert sind:

- ein Klassen- und Methodenbrowser;
- Refactoring-Werkzeuge;
- Objekt-Inspektoren;
- Debugger;
- etc.

Die Umgebung erlaubt die direkte Ausführung von Code mittels der Tastatur und zeigt sofort das Ergebnis.



Der Pharo-Code-Browser

Implementierungen

Es existieren viele Smaltalk-Implementierungen:

**Squeak:** frei verfügbar und plattformunabhängig. Entwickelt von einer aktiven internationalen Gemeinschaft. **VisualWorks:** kommerziell und plattformunabhängig, frei verfügbar für nicht-kommerzielle Nutzung.

**Gemstone:** hochperformante Objektdatenbank. CLASS (Gemstone, Linux, Apache, Seaside, Smaltalk). **Und andere:** GNU Smaltalk, Smaltalk/X, SyX, VA Smaltalk, Dolphin...