

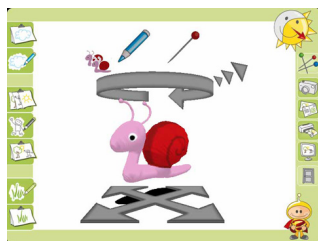
アプリケーション

教育 : EToys (Squeak), SqueakBot, BotsInc, Scratch...

マルチメディア : Sophie, OpenCroquet, Plopp...

Web 開発 : Seaside, Aida, Komanche, Swazoo...

永続性の管理 : オブジェクト指向データベース (Magma, GemStone), 関連データベース (MySQL, PostgreSQL), オブジェクト関連マップ (Glorp).



Plopp でのお絵描き

用語集

イメージ : Smalltalk 環境には、イメージと呼ばれる、システム内での全オブジェクトを永続的に保存できるしくみがあります。イメージは、(クラスやメソッドなどの) オブジェクトとして抽象化したアプリケーションのコードと、状態を含みます。さらに、プログラムの調査やデバッグを行うために使われる開発環境も含まれています。

仮想マシン : 仮想マシンは別のプログラムを実行する能力のあるプログラムです。仮想マシンを使うことで、アプリケーションの移植性を高めるのが容易となります。

リフレクション : 言語がリフレクション能力を持つというのは、プログラムの実行中にコードを調べたり、変更したりする機構があることをいいます。

動的な型付け : 言語の中には、開発者が (整数型や文字列型などの) 変数の型を指定するものがあります。このような言語は、静的に型付けされているといわれます。動的な型付けでは、開発者は変数にどんな型の値を入れても構いません。

書籍

- たくさんのフリーの本一覧:
<http://stephane.ducasse.free.fr/Books.html>
- Smalltalk 全般
 - Smalltalk with Style (Edward Klimas, Suzanne Skublics and David A. Thomas, free)
 - Smalltalk by Example: the Developer's Guide - (Alec Sharp, free)
- Squeak
 - Squeak by Example - (2007, free)
 - Powerful Ideas in the Classroom (BJ Allen-Conn and Kim Rose)
 - 自由自在 Squeak プログラミング (梅澤真史, ISBN:4-88373-203-7, 2004)
 - スクイークであそぼう (Thoru Yamamoto, ISBN:4-7981-0480-9, 2003)

活動

- European Smalltalk User Group conferences (ESUG). 1993 年から活動を開始しました。企業や教育組織の Smalltalker (Smalltalk 利用者) がヨーロッパの国で会合を行なっています。
<http://www.esug.org/conferences>
- 年に 1 度のカンファレンスが、企業や Smalltalk 開発者の協会である STIC (<http://www.stic.st>) の主催で、北米で開かれます。
<http://www.smalltalksolutions.com/>

Internet

- Squeak 公式 website: <http://www.squeak.org>
- 日本の Squeak の拠点: <http://www.squeakland.jp>
- Wiki: <http://wiki.squeak.org>
- ニュース: <http://news.squeak.org>

Smalltalk

純粋な オブジェクト指向で
動的な環境の
プログラミング言語



Smalltalk の重要な概念

Smalltalk は、オブジェクト指向であり、動的に型付けされている、主な文法は 15 分で学習可能な言語です。これは、一貫性に関する以下の事実からきています。

- 全てはオブジェクト: クラスもメソッドも数も、なにかも
- 規則の数はとても少なく、規則の例外はありません

Smalltalk は、仮想マシンを使って実行します。開発は、全ての生きたオブジェクトが含まれ、変更可能なイメージに対して行われます。

Smalltalk の文法

予約語

nil : 未定義オブジェクト (変数のデフォルト値)

true と false : ブール値オブジェクト

self : 現在のオブジェクト

super : スーパクラスコンテキストでの現在のオブジェクト

thisContext : 現在のメソッドの実行時スタック

予約された文字

:= (または ←): 代入

^ (または ↑): メソッドからの結果を返す

| var1 var2 var3 |: 3 つの一時変数の定義

\$a : 文字 a.

#(abc 123) : 二つのリテラル#abc と 123 を含む配列

. : 式の終了

; : メッセージのカスケード

[]: コードブロック (これもオブジェクトです!)

"コメント".

'文字列'.

メッセージの送信

メソッド呼び出しは、メッセージを送信することによって行われます。メッセージ送信の記法は自然言語に基づいており、主語と動詞、および補語から成り立っています。全てのメッセージは、メッセージレシーバと呼ばれるオブジェクトに送られ、オブジェクトを返します。メッセージには、単項、二項、キーワードの 3 種類があります。

単項メッセージ 単項メッセージは引数を取りません。

```
array := Array new.  
array size. ==> 0
```

最初の例は、new メッセージを送信することで、配列 (Array) クラスの新しいインスタンスを作成し、このインスタンスを返します。2 番目の例では、この array の大きさを尋ね、0 が返ってきます。

二項メッセージ 二項メッセージは、引数を一だけ取ります。メッセージ名にはシンボルが使われ、算術式などに使われます。

```
3 + 4. ==> 7  
'Hello', ' World'. ==> 'Hello World'
```

+メッセージがオブジェクト 3 に引数 4 で送られます。後の例では、, メッセージが文字列 'Hello' に引数 'World' で送られます。

キーワードメッセージ キーワードメッセージは、引数を 1 つ以上取ります。引数は、コロンで終わるキーワードの後に書きます。

```
'Smalltalk' allButFirst: 5. ==> 'talk'  
3 to: 10 by: 2. ==> Interval 型で、値は 3,5,7,9
```

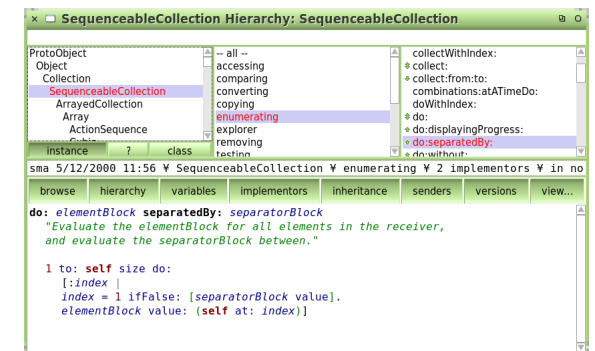
最初の例は、文字列の allButFirst: メソッドを引数 5 で呼び出します。メソッドは、文字列 'talk' を返します。2 番目の例は、要素が 3, 5, 7, 9 のコレクションを返します。

開発環境

Smalltalk の多くは、ソースコードをブラウズしたり、オブジェクトを扱ったりできる統合開発環境 (IDE) と一緒に提供されます。Smalltalk のリフレクション API を使って実装された以下のようなたくさんのツールが利用できます。

- クラスやメソッドブラウザ
- リファクタリングツール
- オブジェクトインスペクタ
- デバッガ
- などなど

このような環境は簡単なキーストロークでコードの評価ができ、すぐにその結果を見ることができます。



Squeak コードブラウザ

実装

いくつかの異なる Smalltalk 実装があります。

Squeak : フリーで、オープンソース、マルチプラットフォームな実装です。国際的なコミュニティによって、活発に開発されています。

Visual Works : プロプライエタリで、マルチプラットフォーム、商用利用しない個人は自由に入手可能です。

Gemstone : プロプライエタリ実装で、とても高性能なオブジェクトデータベースを含みます。

その他 : GNU Smalltalk, Smalltalk/X, SyX, VA Smalltalk, Dolphin...