

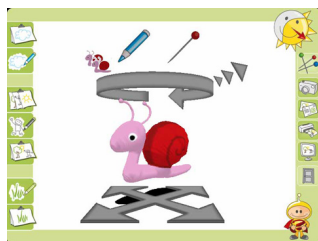
アプリケーション

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 環境は、イメージという永続的なオブジェクトストアを含みます。イメージには、(クラスやメソッドなどの) アプリケーションのコードを含み、オブジェクトはアプリケーションの状態を保持します。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さえ含まれます。

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ある。アプリケーションの移植性を高めるのが容易となる。

リフレクション : 言語がリフレクション能力を持つとは、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 : いくつかの言語は、開発者に各変数の型 (整数や文字列など) を明示することを強制する。これは、静的に型付けされていると呼ばれる。動的な型付けでは、開発者は変数にどんな型の値を入れるのも制限は無い。

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)

活動

- European Smalltalk User Group conferences (ESUG). 1993 年から活動。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う。
<http://www.esug.org/conferences>
- STIC (企業の actors や Smalltalk editors(?) の協会) によって北アメリカで行われている毎年の会議。Annual conference, organised in North America by the STIC, an association with industrial actors and Smalltalk editors.
<http://www.smalltalksolutions.com/>

Internet

- Squeak 公式 website:
<http://www.squeak.org>
- Wiki:
<http://wiki.squeak.org>

Smalltalk

純粹に 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、動的に型付けされている、最小限の文法は 15 分で学習可能な言語です。この主な利点は、以下のとても一貫性のある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、なにもかも
- 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、メッセージの送信で行われます。メッセージは自然言語の方法に習って、The message is based on a natural language, with a subject, a verb and complements. 全てのメッセージはオブジェクトを返します。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ます。単項、二項、キーワードの 3 つの種類のメッセージがあります。

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、array のサイズを尋ね、0 が返ってきます。

二項メッセージ 二項メッセージは、一つだけの引数を取ります。メッセージ名にはシンボルが使われ、しばしば算術式に使われます。

```
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。2 番目の場合、, 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、1 つ以上の引数を取ります。引数は、キーワードの間にはさまれ、キーワードのセミコロンの後に書かれます。

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

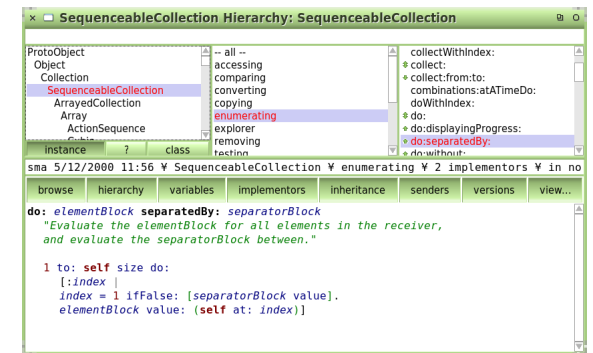
最初の例は、文字列の allButFirst: メソッドを引数 5 で呼び出します。メソッドは、文字列 'talk' を返します。2 番目の例は、要素が 3, 5, 7, 9 のコレクションを返します。

開発環境

多くの Smalltalk 実装は、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。国際的なコミュニティによって、活発に開発されている。

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。

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...