



Trình duyệt mã Pharo

## Phát hành

Có nhiều Smalltalk được phát hành:

Squeak & Pharo: miễn phí, mã nguồn mở và hỗ trợ đa nền. Được phát triển năng động.

VisualWorks: có chủ quyền, hỗ trợ đa nền, miễn phí cho ứng dụng phi thương mại.

Gemstone: phát hành có chủ quyền bao gồm cơ sở dữ liệu đối tượng hiệu quả cao.

Và những thứ khác: GNU Smalltalk, Smalltalk/X, SyX, VA Smalltalk, Dolphin...

## Chương trình ứng dụng

Từ khi được lập ra từ đầu những năm 1980, Smalltalk đã được sử dụng rộng rãi trong nghiên cứu học thuật cũng như trong những ứng dụng thương mại. Sau đây là những ứng dụng Smalltalk tiêu biểu đang được phát triển.

Giảng dạy: EToys (Squeak), SqueakBot, BotsInc, Scratch...

Đa phương tiện (Multimedia): Sophie, OpenCroquet, Plopp...

Phát triển Web: Seaside, Aida, Komanche, Swazoo...

Quản lý lưu trữ: hệ quản lý dữ liệu hướng đối tượng (Magma, GemStone), hệ quản lý dữ liệu quan hệ (MySQL, PostgreSQL), trình ảnh xạ quan hệ đối tượng (Glorp).



Etoys và DrGeo trên máy Mỗi Trẻ Em Một Máy Tính (OLPC)

## Thuật ngữ

Image (Ảnh): Môi trường Smalltalk có một đối tượng để lưu trữ gọi là image (ảnh). Image (ảnh) chứa mã của ứng dụng (gồm các lớp (classes) và các phương thức (methods)), chứa các đối tượng giữ trạng thái của ứng dụng và có thể chứa cả các công cụ phát triển ứng dụng để kiểm tra và tìm lỗi chương trình trong khi nó đang thi hành.

Virtual Machine (Máy ảo): Máy ảo là một chương trình có khả năng thi hành các chương trình khác. Nó làm cho ứng dụng có tính uyển chuyển linh hoạt.

# Smalltalk

ngôn ngữ lập trình  
thuần hướng đối tượng  
và môi trường động



## Những khái niệm quan trọng của Smalltalk

Smalltalk là một ngôn ngữ hướng đối tượng và định kiểu động, với cú pháp đơn giản mà có thể học được trong một vài phút. Lợi thế của nó chính là nhờ vào sự rất chặt chẽ và nhất quán:

- mọi thứ là một đối tượng: các lớp, các phương thức, các số, v.v..
- số qui tắc thì rất ít, và không có ngoại lệ!

Smalltalk chạy trên máy ảo. Sự chặt chẽ trong trình thực hiện trên một image (ảnh) nơi mà tất cả các đối tượng tồn tại và được cấp phát.

## Cú pháp Smalltalk

Các từ danh riêng

<code>nil</code>	đối tượng chưa xác định (giá trị mặc định của biến)
<code>true</code> và <code>false</code>	đối tượng luận lý (boolean)
<code>self</code>	đối tượng hiện tại
<code>super</code>	đối tượng hiện tại trong ngữ cảnh của lớp cha (super class)
<code>thisContext</code>	stack (bộ nhớ chồng) tại thời điểm thực thi của phương thức

Các ký tự danh riêng

<code>=</code> (hoặc <code>+</code> )	gán giá trị
<code>~</code> (hoặc <code>↑</code> )	trả về kết quả của phương thức
<code>  var1 var2 var3</code>	khai báo ba biến tạm
<code>\$a</code>	ký tự a
<code>#(abc 123)</code>	mảng chứa hai phần tử: ký hiệu (symbol) #abc và số 123
<code>.</code> (dấu chấm câu)	kết thúc biểu thức
<code>:</code>	gửi tin noi tiếp (message cascade)
<code>[ ]</code>	Khởi mã (là một đối tượng!)
<code>"ghi chú"</code>	'chúoi ký tự'

Củi tin nhàn

Một phương thức được gọi bằng cách gửi một tin nhàn đến một đối tượng, tin nhàn sẽ trả về một đối tượng.

Tin nhàn được dựa trên ngôn ngữ từ nhien, có chủ ngữ, động từ và bổ ngữ. Có ba loại tin nhàn: nhất ngôn, nhì ngôn và từ khóa.

Tin nhàn nhất ngôn. Tin nhàn nhất ngôn là tin nhàn không có đối số (argument).

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

```
array size.
```

Vi dụ đầu tiên này tạo ra và trả về một bản thể hiện mới của lớp Array (Mảng), bằng cách gửi nó tin nhàn `new`. Vi dụ thứ hai yêu cầu kích thước của mảng này, kết quả là 0.

Tin nhàn nhì ngôn. Tin nhàn nhì ngôn chỉ có một đối số, được đặt tên bằng một ký hiệu và thường được dùng cho các biểu thức toán học.

```
3 + 4.
```

```
'Hello', 'World'.
```

Tin nhàn + được gửi đến đối tượng 3 với 4 là tham số. Trong trình hợp thì hai, tin nhàn , được gửi đến chúoi ki từ 'Hello', với ' World' là tham số.

Tin nhàn từ khóa. Tin nhàn từ khóa có thể có một hay nhiều đối số. Các đối số được chèn vào giữa mọi từ khóa, sau mọi dấu hai chấm.

```
'Smalltalk' allButFirst: 5.
```

```
3 to: 10 by: 2.
```

Vi dụ đầu gửi phương thức `allButFirst`: tren một chúoi ký tự và truyền đối số 5. Trả về chúoi ký tự 'talk'. Vi dụ thứ hai trả về một tập hợp chứa các phần tử là 3, 5, 7 and 9.

Block (Khối)

Các khối là các đối tượng chứa mã không được thực hiện ngay. Chúng là cơ sở cho cấu trúc điều khiển như là điều kiện rẽ nhánh hay điều khiển lặp. Dòng thời, các khối có thể được dùng để gán vào các xử lý tương tác, vi dụ gán vào các mục trình đơn.

## Mọi trường phát triển

Vi dụ này gửi tin nhàn `do`: đến một mảng các chúoi ký tự với một khối là tham số. Khối được tìm giá trị một lần cho mọi phần tử trong mảng. Tham số khối `string` là mọi phần tử của mảng, lần lượt cái này tiếp theo cái khác. Kết quả của toàn biểu thức, lần lượt, theo cái khác. Kết quả của toàn biểu thức, lần lượt, chúoi ký tự 'Hello' , rồi đến 'World' được hiện thì trong cửa sổ theo dõi (Transcript).

```
#('Hello' 'World') do: [:string | Transcript show: string].
```

Hầu hết Smalltalk được cung cấp với một môi trường phát triển tích hợp (IDE), cho phép duyệt mã nguồn và tác động đến các đối tượng. Nhờ vào phần ảnh API, nhiều công cụ được cung cấp sẵn trong Smalltalk như:

- trình duyệt lớp và phương thức (class and method browser);
- công cụ chỉnh sửa cấu trúc nội bộ (refactoring tools);
- trình kiểm tra đối tượng (object inspectors);
- trình bắt sửa lỗi (debugger);
- công cụ quản lý phát hành và điều khiển phiên bản (release management and version control tools);
- và con nhieu ntail

Mã có thể được kiểm tra và định giá trị trực tiếp trong image (ảnh), bằng cách dùng các tổ hợp phím đơn và các trình đơn dạy dù.