

## Chương trình ứng dụng

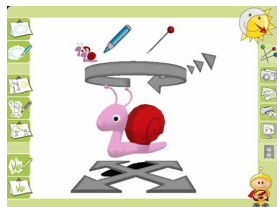
Từ khi được lập ra từ đầu những năm 1980, Smalltalk đã được sử dụng rộng rãi trong nghiên cứu học thuật cũng như trong những ứng dụng thương mại. Sau đây là những ứng dụng Smalltalk tiêu biểu đang được phát triển.

**Giáo dục:** 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).



Vẽ hình bằng Plopp

## 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.

**Reflection (Phản ánh):** Một ngôn ngữ được cho là có tính phản ánh khi mà nó bao gồm cơ chế kiểm tra và cập nhật mã trong khi chương trình đang thi hành.

**Dynamic typing (định kiểu động):** Một số ngôn ngữ buộc lập trình viên định rõ kiểu của từng biến (integer (số nguyên), string (chuỗi ký tự), ...); được gọi là định kiểu cố định. Định kiểu động không buộc theo sự áp đặt này, vì thế làm cho chương trình dễ tái sử dụng hơn và dễ thay đổi hơn.

## Sách

- Nhiều sách miễn phí về Smalltalk có ở đây: <http://stephane.ducasse.free.fr/Books.html>
- Smalltalk đại cương
  - Smalltalk with Style (Edward Klimas, Suzanne Skublics and David A. Thomas, miễn phí)
  - Smalltalk by Example: the Developer's Guide – (Alec Sharp, miễn phí)
- Squeak chi tiết
  - Squeak by Example – (2007, miễn phí)
  - Powerful Ideas in the Classroom (BJ Allen-Conn and Kim Rose)

## Sự kiện

- European Smalltalk User Group conferences (ESUG) (Hội nghị nhóm người dùng Smalltalk châu Âu). Từ năm 1993, Những người dùng Smalltalk (Smalltalkers) trong công nghiệp và trong học thuật tổ chức họp tại châu Âu. <http://www.esug.org/conferences>
- Hội nghị thường niên, được tổ chức ở Bắc Mỹ bởi STIC (<http://www.stic.st>), hội liên hiệp những công ty và những nhà phát triển Smalltalk. <http://www.smalltalksolutions.com/>

## Internet

- Trang web chính thức về Squeak: <http://www.squeak.org>
- Wiki (Thảo luận nhóm): <http://wiki.squeak.org>
- Tin tức: <http://news.squeak.org>

# 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 *mười lăm phút*. Lợi thế của nó chính là nhờ vào sự *rất nhất quán*.

- mỗi 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 phát triển chương trình thực hiện trên một *image (ảnh)*, nơi mà tất cả các đối tượng tồn tại và được cập nhật.

## Cú pháp Smalltalk

### Các từ dành riêng

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

### Các ký tự dành riêng

|                |   |
|----------------|---|
| := (hoặc ←)    | gán giá trị                                   |
| ^ (hoặc ↑)     | trả về kết quả của phương thức                |
| var1 var2 var3 | khai báo ba biến tạm                          |
| \$a            | ký tự a                                       |
| #(abc 123)     | mảng chứa hai phần tử: ký hiệu #abc và số 123 |
| .              | (dấu chấm câu) kết thúc biểu thức             |
| ;              | message cascade (gửi tin nối tiếp)            |
| [ ]            | Khối mã (là một đối tượng!)                   |
| "ghi chú"      |   |
| 'chuỗi ký tự'  |   |

## Gửi tin nhắn

Một phương thức được gọi bằng cách gửi một tin nhắn đến

một đối tượng, tin nhắn sẽ trả về một đối tượng. Tin nhắn được dựa trên ngôn ngữ tự nhiên, có chủ ngữ, động từ và bổ ngữ. Có ba loại tin nhắn: nhất nguyên, nhị nguyên và từ khóa.

**Tin nhắn nhất nguyên.** Tin nhắn nhất nguyên là tin nhắn không có đối số (argument).

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

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

**Tin nhắn nhị nguyên.** Tin nhắn nhị nguyên 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 nhắn **+** được gửi đến đối tượng 3 với 4 là tham số (parameter). Trong trường hợp thứ hai, tin nhắn **,** được gửi đến chuỗi kí tự 'Hello' với 'World' là tham số.

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

```
'Smalltalk' allButFirst: 5.  
3 to: 10 by: 2.
```

Ví dụ đầu gọi phương thức **allButFirst:** trên một chuỗi ký tự và truyền đối số 5. Trả về chuỗi ký tự 'talk'. Ví dụ thứ hai trả về một tập hợp chứa các phần tử là 3, 5, 7 và 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. Đồng thời, các khối có thể được dùng để gắn vào các xử lý tương tác, ví dụ gắn vào các mục trình đơn.

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

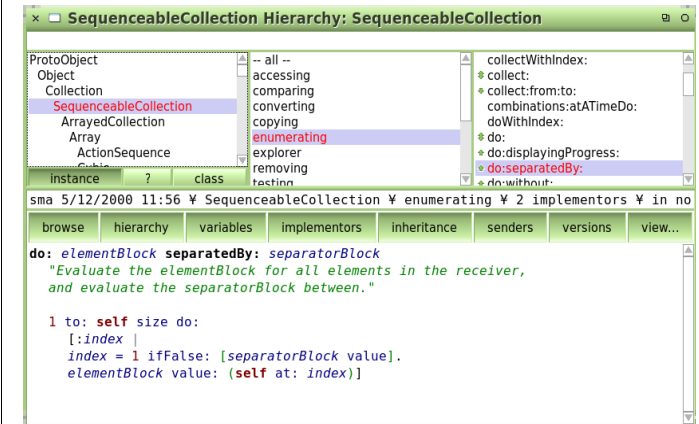
Ví dụ gửi tin nhắn **do:** đến một mảng các chuỗi ký tự với một khối là tham số. Khối được tính giá trị một lần cho mỗi phần tử trong mảng. Tham số khối **string** là mỗi phần tử của mảng, lần lượt cái này tiếp theo cái khác. Kết quả của toàn biểu thức, lần lượt, chuỗi ký tự 'Hello' rồi đến 'World' được trình bày trong cửa sổ theo dõi kết quả (transcript).

## Môi trường phát triển

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 phản ánh API, nhiều công cụ được cung cấp sẵn trong Smalltalk như sau:

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

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 đầy đủ.



Trình duyệt mã Squeak

## Phát hành

Có nhiều Smalltalk được phát hành:

**Squeak:** miễn phí, mã nguồn mở và hỗ trợ đa nền. Được phát triển năng động bởi cộng đồng quốc tế.

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

**Gemston:** 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, ...