

Implémentations

Il existe différentes implémentations de Smalltalk :

Squeak & Pharo : implémentations libres, gratuites et multi-plateformes. Activement développés.

VisualWorks : implémentation propriétaire et multi-plateforme, disponible gratuitement pour une utilisation personnelle.

Gemstone : implémentation propriétaire qui intègre une base de données objet à hautes performances.

Et d'autres : GNU Smalltalk, Smalltalk/X, SyX, VA Smalltalk, Dolphin...

Applications

Éducation : EToys (Squeak pour OLPC), SqueakBot, BotsInc, Scratch, iStoa.net, Edusim, DrGeoII...

Multimédia : Sophie, OpenCroquet, Plopp...

Développement web : Seaside, Aida, Komanche, Swazoo...

Gestion de la persistance : base de données objets (Magma, GemStone), relationnelles (MySQL, PostgreSQL), mapping objet relationnel (Glorp...).



Etoys et DrGeo sur un OLPC

Glossaire

Image mémoire : L'environnement Smalltalk contient un entrepôt persistant d'objets, l'image. Elle contient le code de l'application (les classes et les méthodes), les objets qui constituent l'état de l'application et peut même contenir l'environnement de programmation pour inspecter et déboguer le programme pendant qu'il s'exécute.

Machine virtuelle : Une machine virtuelle est un programme qui est capable d'exécuter d'autres programmes. Cela permet de faciliter la portabilité des applications que l'on développe.

Réflexion : On dit qu'un langage est réflexif s'il dispose de mécanismes permettant d'inspecter et de modifier du code pendant l'exécution d'un programme.

Smalltalk

un langage de programmation
purement orienté objet
et un environnement dynamique



Smaltalk est un langage orienté objet, à typage dynamique dont la syntaxe est minimale et peut s'apprendre en quinze minutes.

Smalltalk fonctionne sur le principe d'une machine virtuelle. Le développement se fait dans une image mémoire dans laquelle se trouve l'ensemble des objets du système avec lesquels il est possible d'interagir.

Mots réservés

Caractères réservés	:= (no +)	affectation
---------------------	-----------	-------------

portales
\$a le caractère a
#(abc 123) tableau contenant deux litté-
raux : le symbole #abc et le

COINTEGRATION

L'appel de méthode se fait par envoi de message. Le message se construit sur la base du langage naturel avec un sujet, un verbe et des compléments. Tout envoi de message retourne un objet. Tous les messages sont envoyés à un objet que l'on appelle le receveur du message. Il existe trois types de messages : unaire, binaire et à mots-clés.

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

Messages binaires. Un message binaire ne prend qu'un argument, se nomme par un symbole et est souvent utilisé pour des opérations arithmétiques.

Le message + est envoyé à l'objet 3 avec comme para-

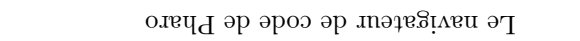
Messages à mots-clés. Un message à mots-clés peut prendre un ou plusieurs arguments. Les arguments sont insérés entre chaque mot-clé, après les deux-points.

Le premier exemple appelle la méthode `allButFirst`:

Les blocs sont des objets qui contiennent du code non exécuté immédiatement. Ils sont à la base des structures de contrôles comme les conditionnelles et les boucles.

Environnement de développement

- un navigateur de classes et de méthodes (+ + + relacto-
risation);
- un inspecteur d'objets ;
- un débogueur ;
- et beaucoup plus ;

[illegible]

Le navigateur de code de Pharo