



## Comment les implémenteurs influencent la programmation en Pharo

Les « implémenteurs » (« implementors ») influencent profondément la programmation en Pharo en facilitant l'exploration, la compréhension et la modification du code dans un environnement interactif et vivant. Cette fonctionnalité est un pilier du workflow itératif typique de Pharo et Smalltalk. <sup>[1]</sup>

### Navigation et compréhension du code

La possibilité de lister instantanément toutes les classes qui définissent une méthode donnée permet de rapidement cerner comment un comportement est distribué, spécialisé ou redéfini dans la hiérarchie des classes. Cela accélère l'apprentissage d'un nouveau codebase et réduit le coût cognitif de la maintenance ou de l'évolution du logiciel. <sup>[1]</sup>

### Refonte et extension du système

Les implémenteurs rendent naturelles les étapes de refactoring et d'extension : retrouver tous les points d'implémentation d'une méthode permet de systématiquement appliquer ou corriger un comportement partout où ce message existe, garantissant ainsi cohérence et robustesse dans le système. <sup>[1]</sup>

### Programmation exploratoire et dynamique

Dans le paradigme Pharo, où tout est objet et où le système est toujours vivant, la facilité d'accès aux implémenteurs renforce un style de programmation exploratoire. Le développeur peut interagir directement avec son environnement, naviguer entre concepts et comportements, puis itérer immédiatement sur son code, tout en visualisant facilement l'impact de ses modifications à chaque niveau. <sup>[1]</sup>

En résumé, les implémenteurs permettent une navigation efficace et une compréhension fine du système, et s'intègrent au cœur du workflow « vivant » de Pharo, où explorer, modifier, puis ré-exécuter sont autant d'actions immédiates et naturelles. <sup>[1]</sup>



1. <https://pharo.org/features>
2. <https://rmod-files.lille.inria.fr/Team/Texts/Papers/Thoma24b-Cola-Reflective-Official.pdf>
3. <https://files.pharo.org/books-pdfs/pharo-by-example/2011-haroByExample-2011-FR.pdf>
4. <https://connect.ed-diamond.com/GNU-Linux-Magazine/glmf-189/un-chat-en-pharo-le-serveur>

5. <https://files.pharo.org/books-pdfs/updated-pharo-by-example/2016-10-27-UpdatedPharoByExample.pdf>
6. <https://linuxfr.org/news/pharo-quoi-de-neuf>
7. <https://www.youtube.com/watch?v=wo5YAYZ0v0w>
8. <https://thepharo.dev/category/data-structures/>
9. [https://www.univ-lille.fr/fileadmin/user\\_upload/Recherche/Plateformes\\_et\\_grands\\_equipements/Fiches\\_marketing/Pharo.pdf](https://www.univ-lille.fr/fileadmin/user_upload/Recherche/Plateformes_et_grands_equipements/Fiches_marketing/Pharo.pdf)
10. <https://www.fun-mooc.fr/fr/cours/programmation-objet-immersive-en-pharo/>
11. <https://github.com/pharo-open-documentation/awesome-pharo>
12. <https://www.fun-mooc.fr/fr/actualites/5-raisons-de-suivre-le-mooc-programmation-objet-immersive-en-pharo/>
13. <https://linuxfr.org/news/sortie-du-langage-pharo-et-de-son-environnement-de-developpement-en-version-3-0>
14. <https://programmation.developpez.com/actu/344537/La-version-11-de-Pharo-le-langage-de-programmation-et-environnement-d-execution-purement-oriente-objet-est-disponible-avec-plusieurs-ameliorations-au-niveau-de-la-convivialite-et-de-la-rapidite/>
15. <https://connect.ed-diamond.com/GNU-Linux-Magazine/glmf-123/optimisation-d-applications-en-pharo>
16. <https://programmation.developpez.com/tutoriel/cours-complet-pharo/?page=chapitre-5-moins-le-modele-objet-de-smalltalk>
17. <https://www.cherry.py.org/smalltalk-linfluence-dun-langage-oublie-sur-la-programmation-orientee-objet/>
18. <https://programmation.developpez.com/tutoriel/cours-complet-pharo/?page=chapitre-1-moins-une-visite-de-pharo>
19. <https://inria.fr/fr/le-langage-pharo-fete-ses-10-ans-apres-un-tour-du-monde>
20. <https://www.lirmm.fr/~dony/enseig/MR/TD-HAI931-245.pdf>