

Wollok – y una frase que describa la idea

Nicolás Passerini

Javier Fernandes

October 13, 2014

Abstract

In this context... We consider this problem P... P is a problem because... We propose this solution... Our solution solves P in such and such way.

1 Introduction

(a) hay que arrancar introduciendo nuestra visión, contando por qué enseñamos objetos de determinada manera, por qué hicimos Ozono.

Enseñanza tradicional: - Muchas veces se parte del lenguaje - Bajos niveles de aprobación - Pasa mucho tiempo hasta que un chico puede hacer un programa "real", demasiados conceptos. - ... después pienso más

Nosotros propusimos: - Elegir un recorrido que permite ir incorporando los conceptos uno a uno - Tener una herramienta que da soporte a eso - Focalizar en objeto-mensaje-polimorfismo-referencias, los demás conceptos aparecen después.

Also these hindrances reduce the opportunity of students to apply the concepts of the paradigm effectively in their further professional practice, resulting in several IT-projects not taking advantage of the possibilities offered by the potential of good object-oriented practices, even in cases where the tools used may allow the application of object-oriented programming gracefully. [2]

We propose to provide the student a reduced and graphical programming environment in which the object and the message are the central concepts instead of defining classes and then instantiate them. [1]

(b) lo que aprendimos en estos 8 años haciendo eso, que nos lleva a querer darle una vuelta más.

Si bien todo esto que se nos ocurrió en su momento fue maravilloso y genial y los pibes aprenden más, programan mejor y la ponen más seguido, fuimos aprendiendo más cosas y estamos trabajando en unos cambios:

Los fáciles: - integrar clases y objetos en un mismo programa - integrar interfaces de usuario automáticas - utilizar herramientas avanzadas para guiar a los alumnos en el proceso de aprendizaje, la vedette acá sería el sistema de tipos. - acercar la práctica de lo que hacemos a la práctica industrial (acá hay que ver qué decimos y qué no, por ejemplo me molesta que no sea un archivo... pero desde lo metodológico se puede

pensar en tests o incluso en un repositorio de código... sobre la relación con la industria se podría escribir un libro, hay que ver cuánto queremos meternos).

No sé si hablar de los temas de colecciones, son un poco particulares de Ozono.

(c) las ideas nuevas... y wollock como herramienta para dar soporte a esas ideas.

Lo concreto que hicimos es tener un lenguaje con - clases y objetos integrados - un IDE profesional con syntax highlighting, refactors, autocompletion y la vedette (?) inferencia de tipos (en progreso). - bueno, muchas mejoras a nivel lenguaje, como literales para colecciones, imports - una forma fácil de construir tests.

2 Problem Description

Context, exposed with the **most precise terms possible** (don't open unwanted doors for the reader)

Probably set the vocabulary before to cut any misinterpretation

Constraints that influenced the solution (because the solution is not universal) *e.g.* our requirements for a solution, possibly not all satisfied. They should be sound and believable. Analysis of the criteria. Imagine that you are another guy having this problem do the constraint matches yours so that you could apply the solution

3 Proposed Solution

4 Discussion

5 Related Works

6 Conclusion

Y a futuro le agregaríamos: - Integración con la UI - Integración fácil con SCM - Más refactors y mejora del sistema de inferencia. - Una versión web / versión liviana con el objetivo de poder ejecutarse en las netbooks que tienen los chicos de secundaria.

Probar la nueva herramienta en entornos educativos.

Eso pensando el lenguaje/herramienta, si pienso a nivel docencia se me ocurre que lo que tenemos que hacer es integrarnos con otras entidades, como Sadosky u otras universidades. Haciendo foco en que el punto no es la herramienta sino que tenemos que repensar cómo enseñamos a programar.

Acknowledgements

References

- [1] C. Griggio, G. Leiva, G. Polito, G. Decuzzi, and N. Passerini. A programming environment supporting a prototype-based introduction to OOP. In *Proceedings of*

the International Workshop on Smalltalk Technologies, IWST '11, pages 5:1–5:5, New York, NY, USA, 2011. ACM.

- [2] C. Lombardi, N. Passerini, and L. Cesario. Instances and classes in the introduction of object oriented programming. DC - FCEN - UBA, Buenos Aires, Argentina, Dec. 2007.