Object-Oriented Programming

Objects and Design in Smalltalk

Éric Tanter
etanter@dcc.uchile.cl





"[programming languages are] either an agglutination of features or a crystalization of style."

"COBOL, Ada, [Java], etc., belong to the first kind; LISP and Smalltalk are the second kind."

- Alan C. Kay, 1993

"A language that does not affect the way you think about programming is not worth knowing."

- Alan Perlis, 1982

"One can't talk about objects and Smalltalk without also talking in practically the same breath about design."

"[...] to design is to discover alternatives, weigh them, and consciously choose among them."

"Design, like life, is all about striking the right balance."

- Chamond Liu
"Smalltalk, Objects and Design"
2nd Edition, 2000

Language Design

- Yardsticks of language design
  - orthogonality: keep unrelated features unrelated
  - generality: use an operation for many purpose
  - parsimony: delete unneeded operations
  - completeness: can the language describe all objects of interest?
  - similarity: make the language as suggestive as possible
  - extensibility: make sure the language can grow
  - openness: let the user "escape" to use related tools

Jon Bentley, "Programming Pearls - Little Languages", Communications of the ACM 1986, 29(8)

Why Smalltalk?

- Purist approach: everything is an object
- Uniform, simple, powerful
  - Dynamically typed
  - No primitive types! (everything...)
- Simple syntax
  - Forced to think
  - Discuss OO design and modeling, not syntax

Why Smalltalk?

- Object culture
- Environment completely written in Smalltalk (everything...)
- Powerful development tools
  - Refactoring browser, SUnit, code versioning, etc.
- Industrial applications
  - UPS Packet Tracking system, JPMorgan KAPITAL system (risk management), AMD, ...
  - web servers, wikis, etc.

Squeak?

- Pros
  - developed by the inventors of Smalltalk
  - open source
  - fun: multimedia, 3D, eToys...
  - used for teaching maths/science to kids
- Cons
  - you may think that Smalltalk is not serious
  - code is not always very good

VisualWorks?

- Cons
  - not open source
  - less fun that Squeak
- Pros
  - industry-oriented
  - professional tools
  - Store/RB
  - code is better
  - non-commercial version for free

Credits

- The material of this lecture is essentially based on the -excellent- book by Chamond Liu
  - "Smalltalk, Objects, and Design", iUniverse.com, 2000
  - and the courses of Stéphane Ducasse
- Material donated to U.Chile by ESUG
  - www.esug.org

References

- Google!
- Squeak
  - www.squeak.org
  - www.mucow.com/squeak-qref.html
- VisualWorks/Cincom
  - http://smalltalk.cincom.com
- Smalltalk for PDAs:
  - www.pocketsmalltalk.com/
- Free Books!
  - www.iam.unibe.ch/~ducasse/FreeBooks.html

Challenges

- There is much more to OOP than C++/Java
  - get to know Smalltalk, Squeak, etc.
  - change your mind for a different world
  - give you the taste for language design
    - there is more to OOP that class-based languages...
- Object-oriented thinking
  - this is the hardest part
  - programming with an OO language DOES NOT mean programming OO!