Advanced Programming

António Menezes Leitão

February 14, 2024

Advanced Programming

Definition

- Non-trivial programming?

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession
  - Meta-programming

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession
  - Meta-programming
  - Macros

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession
  - Meta-programming
  - Macros
  - Multiple dispatch

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession
  - Meta-programming
  - Macros
  - Multiple dispatch
  - Method combination

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession
  - Meta-programming
  - Macros
  - Multiple dispatch
  - Method combination
  - Multiple inheritance/Mixins

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession
  - Meta-programming
  - Macros
  - Multiple dispatch
  - Method combination
  - Multiple inheritance/Mixins
  - Lazy evaluation

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession
  - Meta-programming
  - Macros
  - Multiple dispatch
  - Method combination
  - Multiple inheritance/Mixins
  - Lazy evaluation
  - Non-deterministic evaluation

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession
  - Meta-programming
  - Macros
  - Multiple dispatch
  - Method combination
  - Multiple inheritance/Mixins
  - Lazy evaluation
  - Non-deterministic evaluation
  - Meta-circular evaluators

Advanced Programming

Definition

- Non-trivial programming?
- Programming using non-trivial languages?
- Programming using non-trivial concepts?
  - Introspection/Intercession
  - Meta-programming
  - Macros
  - Multiple dispatch
  - Method combination
  - Multiple inheritance/Mixins
  - Lazy evaluation
  - Non-deterministic evaluation
  - Meta-circular evaluators
- Programming without ChatGPT?

Example: The Julia Language (2012)

- MIT's language for scientific computing
- Dynamic type system
- First-class types
- Multiple dispatch
- Lisp-like macros and other metaprogramming facilities
- Designed for parallelism and distributed computation
- Coroutines
- User-defined types are as fast and compact as built-ins
- Automatic generation of efficient, specialized code
- Read-eval-print-loop

Example: The Verse Language (2022)

- Epic Games' language for the Metaverse
- Static type system
- First-class types
- Functional programming
- Logic programming
- Existential variables
- Unification
- Multiple values
- Transactional
- Lenient evaluation

Bibliography

Papers, provided for each topic, e.g.:

Jason Baker and Wilson C. Hsieh.

Maya: Multiple-dispatch syntax extension in java.
In *PLDI*, pages 270–281, 2002.

S. Chiba.

Javassist – A reflection-based programming wizard for java.
In *Proceedings of the Workshop on Reflective Programming in C++ at the 13th ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'98)*, Vancouver, Canada, October 1998.
http://www.csg.is.titech.ac.jp/∼chiba/oopsla98/proc/chiba.pdf.

Shigeru Chiba.

Load-time structural reflection in Java.
Lecture Notes in Computer Science, 1850:313–??, 2000.

Sheng Liang and Gilad Bracha.

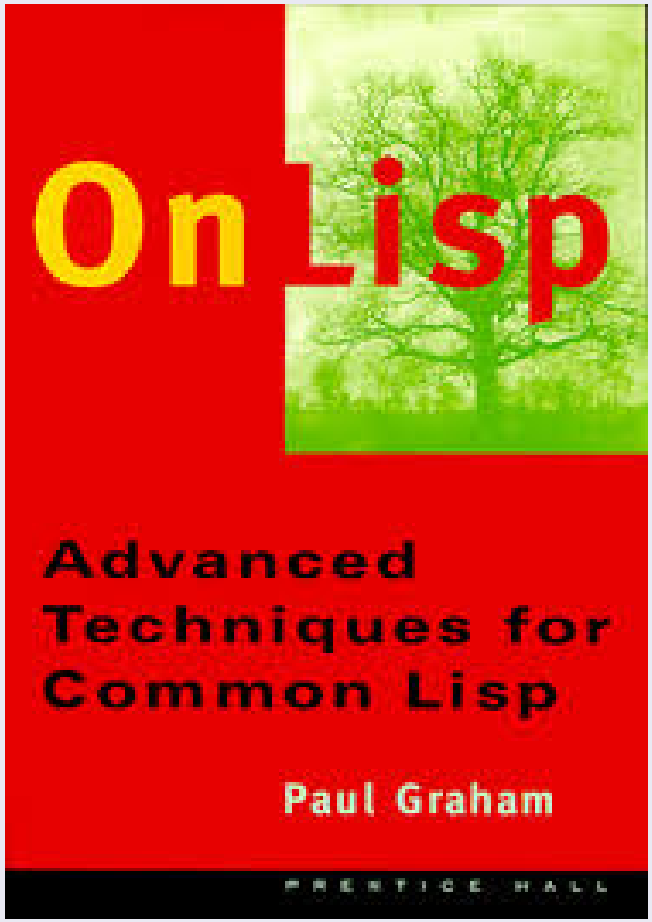
Dynamic class loading in the java tm virtual machine.
In *In Proc. 13th ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'98), volume 33, number 10 of ACM SIGPLAN Notices*, pages 36–44. ACM Press, 1998.

Pattie Maes.

Concepts and experiments in computational reflection.
In Norman Meyrowitz, editor, *Proceedings of the 2nd Annual Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA '87)*, pages 147–155, Orlando, FL, USA, October 1987. ACM Press.

Bibliography

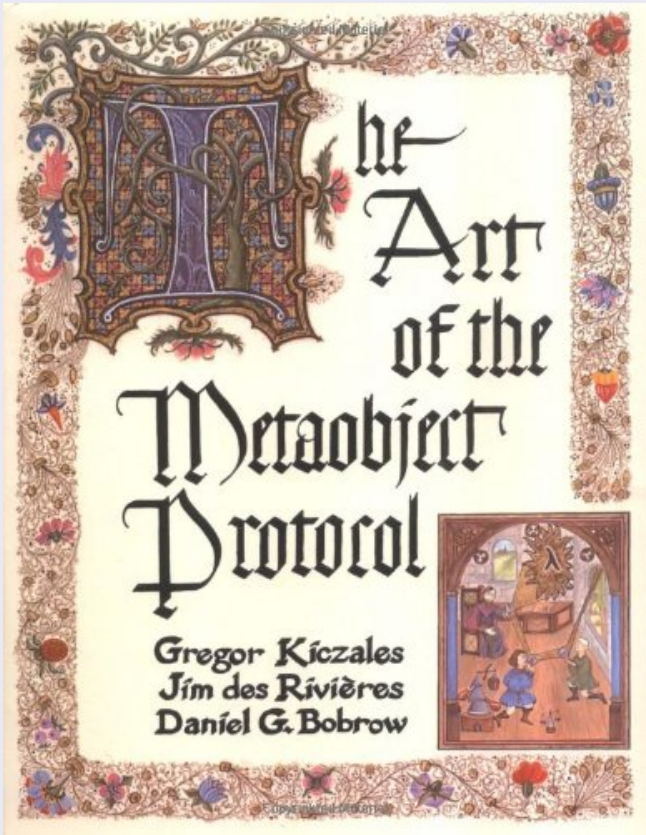
Books, e.g.:



On Lisp: Advanced Techniques for Common Lisp, Paul Graham, Prentice Hall, 1993 (freely available).

Bibliography

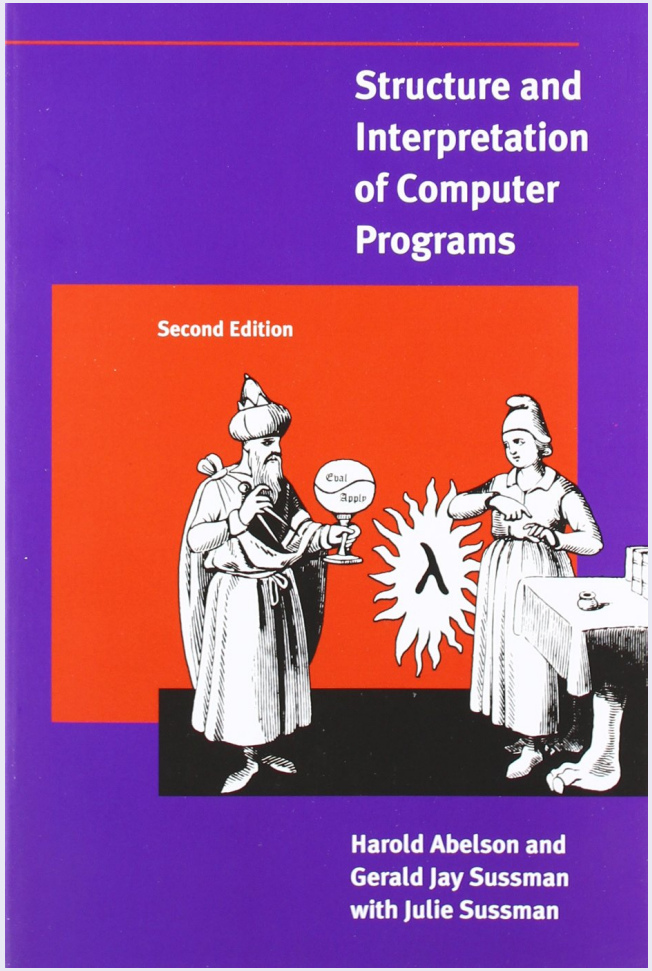
Books, e.g.:



The Art of the Metaobject Protocol, Gregor Kiczales, Jim des Rivieres, and Daniel G. Bobrow, MIT Press, 1991.

Bibliography

Books, e.g.:



Structure and Interpretation of Computer Programs, Gerald Jay Sussman and Hal Abelson, MIT Press, 1996 (freely available).

Evaluation

Project

- One project
- 60%
- Groups of **three** or **four** students
- Weeks: 4, 5, 6, 7.
- With extensions, highest possible grade is 22.

Exam

- One exam
- 40%
- 2 hours

Projects

Examples

- Java Object Inspector
- Method Combination in Java
- AutoLisp in Scheme
- REPL for Java
- Common Lisp - Java Connection
- Scheme in Java
- Extensions to Java's Type System
- Multi-Methods in Racket
- Tracer for Java
- Debugger for Java
- Autoboxing profiler for Java
- Exception-handling in Julia
- Multiple-dispatch for Java
- Method combination in Julia
- Meta-Object Protocol for Julia

Projects

The singularity is near!

- An idea first proposed by Vernor Vinge in 1993 and popularized by Ray Kurzweil in 2005.
- When the development of AI reaches a point where it can improve and evolve on its own, without human intervention.
- This may lead to an acceleration in technological progress, leading to unprecedented and unpredictable changes in society.
- It already affected farmers, tourism operators, factory workers, etc, and is now affecting teachers, students, and programmers.

Projects

The singularity is near!

- An idea first proposed by Vernor Vinge in 1993 and popularized by Ray Kurzweil in 2005.
- When the development of AI reaches a point where it can improve and evolve on its own, without human intervention.
- This may lead to an acceleration in technological progress, leading to unprecedented and unpredictable changes in society.
- It already affected farmers, tourism operators, factory workers, etc, and is now affecting teachers, students, and programmers.

Yeah, yeah, but

- Can I use ChatGPT/Copilot/Tabnine/Ghostwriter/etc to help me implement the PAva project?

Projects

The singularity is near!

- An idea first proposed by Vernor Vinge in 1993 and popularized by Ray Kurzweil in 2005.
- When the development of AI reaches a point where it can improve and evolve on its own, without human intervention.
- This may lead to an acceleration in technological progress, leading to unprecedented and unpredictable changes in society.
- It already affected farmers, tourism operators, factory workers, etc, and is now affecting teachers, students, and programmers.

Yeah, yeah, but

- Can I use ChatGPT/Copilot/Tabnine/Ghostwriter/etc to help me implement the PAva project?
- Yes, but be careful. Always check IST's *Code of Conduct*.

Finally

Questions?