An Overview of the Java Programming Language

(Preview for the 9/15 meeting)

Zhong Shao

Department of Computer Science
Yale University

September 1999

---

Reading list

- **Required:**
  - LY99 Chap 2 Java PL Concepts (handout; also online)
  - Bill Pugh's Java crash course slides (on the web)

- **References:**
  - AG97: The Java Programming Language (2nd Edition)
  - GJS96: The Java Language Specification (online)
  - Java spec report (a web site discussing dirty issues in Java)
  - Java programmer's FAQ (available online)

- **If you are unfamiliar with Java:**
  - read AG97 or David Flanagan's "Java in a Nutshell" and "Java Examples in a Nutshell" or Pugh's "Learning Java" web site.
Objectives

- A thorough understanding of important Java features
  - Focus on their specifications & rules rather than how to program with them
  - Goal: create a Java language specification “cheat-sheet”

- Comparison with other languages (ML, Haskell, C, C++)
  - what a modern programming language should support?
  - what are Java’s strong points?
  - what are Java’s weak aspects?

- Create a list of topics that we should revisit
- Create a list of research problems on the Java language design and extension

What I will cover?

- Big picture
- Java basics
  - unicode; types, values, and variables
  - conversions and promotions
  - names and packages
- OO support
  - classes; interfaces; constructors; methods; fields; initializers
  - the (modifiers x features) matrix
  - how to program with interfaces?
- Arrays and exceptions
- Execution basics
- Java reflection (if time permits)
What I will NOT cover?

- Statements and expressions syntax (Oct 19, 26)
- Inner classes (probably a future topic)
- Execution details (what this whole seminar is about)
- Concurrency features (Nov 3)
- Documentation comments
- Java coding conventions
- Java class libraries
- Binary compatibility (Dec 8)

Things I want to find out

- Method invocation and field access on the "null" value?
- Method overriding in the presence of the throw clause?
- How the throw clause is checked?
- What are marker interfaces?
- How the Array class differs from other classes
- The Java exception hierarchy; is it much better than exceptions in ML or Modula-3?
- How does the initializer work?
What I may spend more time on?

- OO features
- Arrays
- Exceptions
- Classes Object and Class
- reflection features

Potential topics for discussion

- What features in Java we really like and what we really hate?
- Comparison with other programming languages
- What a modern prog. language should support?
- What are interesting language extensions over Java?