MeetUp

Virtual MeetUp Tuesday April 7

Time: 10:00 AM Pacific / 5:00 PM GMT

Location: Lucy's dedicated Google Hangout: http://s.apache.org/lucy-hangout.

To participate, log into Google Plus, then join the videochat.

If you're having difficulty joining the Hangout, find us at the `#lucy_dev` IRC channel on irc.freenode.net.

Agenda

1. Introductions (10 minutes) 2. Interfaces (30 minutes) 3. Open discussion / more about interfaces (20 minutes)

Format

This MeetUp will take the form of a Lucy Book Club discussion: participants will answer questions prepared by other participants on the theme. Feel free to bring your own questions!

General discussion questions for interfaces:

1. It is impossible to add a method a Go interface without breaking backwards compatibility. Same with Java interfaces prior to Java 8. Why?
2. Why is it possible to add methods to an abstract class without breaking backwards compatibility?
3. What is a "default method" in Java 8? How do default methods make it possible to modify a Java interface without breaking backwards compatibility?
4. How might you emulate abstract methods in a dynamic programming language?
5. How might you emulate interfaces (without default methods) in a dynamic programming language?
6. How might you emulate interfaces (with default methods) in a dynamic programming language?
7. What's the difference between a "mixin" and an interface consisting entirely of default methods (i.e. with no abstract methods)?

Clownfish/Lucy-specific discussion questions for interfaces:

1. Should Lucy's Query class be an interface instead of an abstract class? How about Analyzer, Searcher, Matcher, Collector, Compiler, IndexReader, DataReader, DataWriter, Lexicon, Posting, Folder, FileHandle, Lock, and SortExternal – all of which are abstract classes?
2. There are a handful of Lucy classes which are designed to be subclassed by users but are not abstract: QueryParser, IndexManager, Schema, FieldType, and Highlighter. Why is it challenging to change these to interfaces?
3. How can we change the Clownfish callback mechanism to be based on interfaces instead of classes?
4. Should Obj#Hash_Sum be moved to a "Hasher" interface? Should Obj#Clone be moved to a "Cloner" interface? Should Obj have any methods at all?

Bonus questions:

1. What are covariant return types?
2. Why is duck typing easier using hashtable-based method dispatch than vtable-based method dispatch?

Optional reading materials:

- Mixins
- Default methods