Grace
Manifesto for a New Educational Object-Oriented Programming Language

Andrew Black, Kim Bruce, James Noble

Supporters

Peter Andreae, Victoria University of Wellington
Gilad Bracha, Ministry of Truth
John Boyland, University of Wisconsin, Milwaukee
Pascal Constanza, Vrije Universiteit Brussel
Sophia Drossopoulou, Imperial College, London
Susan Eisenbach, Imperial College, London
Michael Hicks, University of Maryland
Michael Kölling, University of Kent at Canterbury
Gary Leavens, University of Central Florida
Shane Markstrum, Bucknell University
Doug Lea, SUNY Oswego
Dirk Riehle, Friedrich-Alexander-University of Erlangen-Nürnberg
Ewan Tempesta, The University of Auckland
Dave Thomas, Bedarra Research Labs
Laurence Trott, Middlesex University
Jan Vitek, Purdue University
What is an educational programming language?

- Designed specifically for novices
- Can have limited or broad domain of application
  - We are interested in broad domain
- Main focus is on programming in the small, but some modularity features.

Teach Industrial-Strength Languages?

- Too much conceptual redundancy
- High overhead for simple programs
  - Too hard to read and write
- Want clean concepts
- Saddled w/backward compatibility

What Makes a Language Succeed?

- Clean, simple design
  - good enough, perfection not required
- Widely available
- Timing – dissatisfaction with alternatives
  - Java succeeded in part because of unhappiness with C++
  - Blue suffered
- Support movement of new ideas to intro level
  - Pascal: top-down design, structured programming

User Model

- First year students in OO CS1 or CS2
  - objects early/late, static/dynamic, functional/procedural/scripting...
- Second year students
- Faculty & TAs — assignments and libraries
We are in the dog food business

User model:
Beginning students

Customer:
experienced instructors

The consumer is not the customer

Principles & Motivations
James

Why Now?
- Happy teaching Java next 3-5 years
- In 2015, Java will be 20 years old
- State of the art has advanced
  - patches look like patches
- New languages bring great ideas
- But are for professionals, not students
- To be ready in 2015, we need to start now.

Motivations
- In early part of curriculum want
  - Low overhead for simple programs
  - Language levels
  - Solid generics
  - Static and dynamic typing
  - High level constructs for concurrency/parallelism
  - Support for immutables
  - Power of functional constructs
**Grace Fundamentals**

- Everything is an object
- Simple method dispatch
- Single inheritance
- Language levels for teaching
- Extensible via Libraries (control & data)
- Java / C / Python / Scala programmers should be able to read Grace programs

---

**Features**

- Uncluttered code; layout significant
- Structural typing
- Local type inference
- Separate subtyping from inheritance
- User-definable operators
- Sensible generics
- Lambdas
- Supports static and dynamic typing
- Parallel programming
- Equals & hashcode work automatically
- v instead of getV() for access
- Minimize incantations (public static void main)

---

**What Next?**

Andrew

---

**Process**

- Openness
- 3 Rules of 3
  - 3 compelling examples
  - 3 existing languages
  - We 3 decide!
- Make our own mistakes not anyone else's
Blog

Material for review will be available at http://gracelang.org/
Not much there yet.

Help!

- Supporters
- Programmers
- Implementers
- Library Writers
- IDE Writers
- Testers
- Teachers
- Students
- Tech Writers
- Textbook Authors
- Blog editors
- Community Builders

How to Get Involved

- Subscribe to RSS feed at gracelang.org
- Watch for e-mail list