

มโนทัศน์ภาษาโปรแกรม

แนะนำวิชา

แผนการสำหรับวันนี้

- เป้าหมาย/ความสำคัญ
- เนื้อหาโดยรวม
- รายละเอียดวิชา
- ประวัติศาสตร์ (ซ้ำรอย?)
- เรื่องเด่น

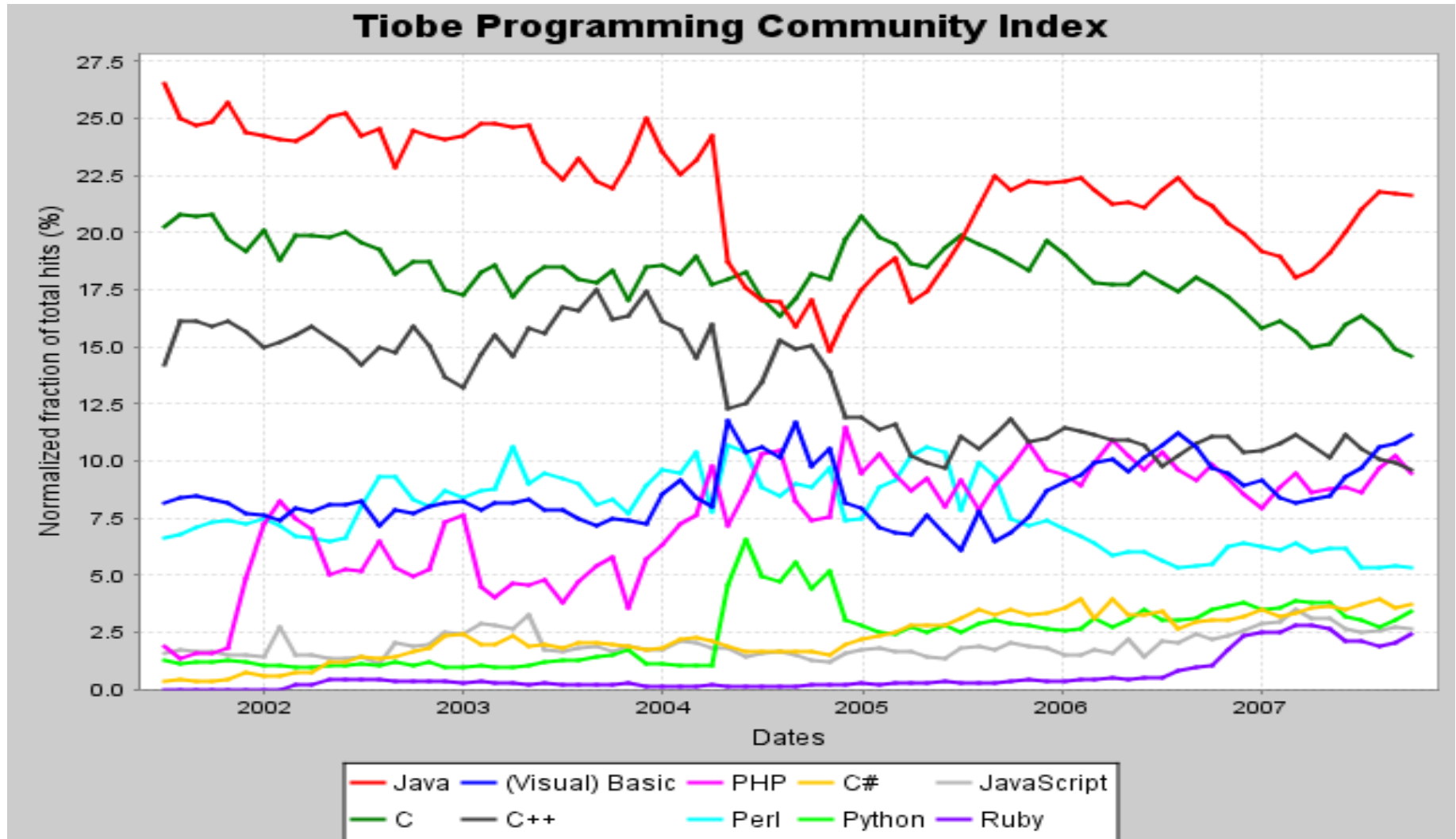
ภาษาโปรแกรม

- ภาษาเป็น “กรอบความคิด”
- ภาษาโปรแกรมเป็น “กรอบความคิด” ที่เราใช้ในการโปรแกรม
 - มีหลากหลาย
 - น่าสนใจ!
 - น่าทึ่ง!
- ภาษาโปรแกรม = สื่อกลางในการอธิบายการคำนวณ

เป้าหมาย

- ศึกษากระบวนการทัศน์ต่าง ๆ ในภาษาโปรแกรม
 - เข้าใจและเปรียบเทียบได้
 - จำเป็นอย่างยิ่ง เนื่องจากปัจจุบันมีการนำกระบวนการทัศน์ต่าง ๆ มาผสมผสานกัน
- เห็นภาพของการออกแบบภาษา
 - การออกแบบคือเรื่องของการ trade-off
 - เรียนรู้เกี่ยวกับประวัติศาสตร์และที่มาของทางเลือกที่ใช้ในการออกแบบ
- เข้าใจ implementation ของคุณสมบัติต่าง ๆ

เทรนด์ภาษาโปรแกรม



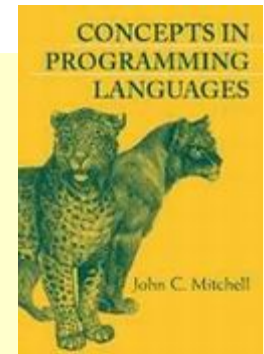
- จาก <http://www.tiobe.com/tpci.htm>

ความเร็วของภาษา

- เราจะวัดความเร็วของภาษาได้อย่างไร?
 - เราวัดไม่ได้ เราวัดได้แต่ความเร็วของ implementation ของแต่ละภาษา
- เราจะวัดความเร็วของ implementation ได้อย่างไร?
 - เราวัดไม่ได้ เราวัดได้แต่ความเร็วของโปรแกรมที่เขียน
- ดูตารางได้จาก The Computer Benchmarks Game: <http://shootout.alioth.debian.org/>

เนื้อหาโดยรวม (ครั้งแรก)

- พื้นฐาน
 - ภาษาโปรแกรมเชิงฟังก์ชัน Lisp/Scheme
 - แลมบ์ดาแคลคูลัส
- ระบบชนิดข้อมูลและการอนุมานชนิด (Type system/type inference)
- โครงสร้างแบบบล็อกและการจัดการหน่วยความจำ (Block structure and memory management)
- โครงสร้างควบคุม (Control structure)
- ภาษาเชิงสคริปต์ (script languages)
/ภาษาโปรแกรมพลวัต (dynamic languages)



เนื้อหาโดยรวม (ครั้งแรก) -- ตามภาษา

- พื้นฐาน
- ระบบชนิดข้อมูล
- โครงสร้างแบบบล็อกและ
การจัดการหน่วยความจำ
- โครงสร้างควบคุม
- ภาษาเชิงสคริปต์
/ภาษาโปรแกรมพลวัต
- Scheme
- ML
- Ruby/Python

เนื้อหาโดยรวม (ครึ่งหลัง)

- โมดูล ข้อมูลแบบนามธรรม วัตถุ (Modularity, data abstraction, objects)
- ภาษาโปรแกรมเชิงวัตถุ (Object-oriented languages)
- ?? ภาษาโปรแกรมเชิงตรรกศาสตร์ (*Login programming*)
- กรอบงานการโปรแกรม (Programming frameworks)
- เทคโนโลยีการโปรแกรมอื่น ๆ

เนื้อหาโดยรวม (ครึ่งหลัง) -- ตามภาษา

- โมดูล ข้อมูลแบบนามธรรม วัตถุ
- ภาษาโปรแกรมเชิงวัตถุ
- ML
- Smalltalk
- C++
- Java / C#
- Ruby / Python
- ?? ภาษาโปรแกรมเชิงตรรกศาสตร์
- Prolog
- Ruby on rails / Django
- กรอบงานการโปรแกรม

การสัมผัสโดยตรง

- โดยทั่วไปแล้วการศึกษามโนทัศน์ภาษาโปรแกรมไม่จำเป็นต้องเขียนโปรแกรมภาษาต่าง ๆ ก็ได้
- อย่างไรก็ตาม จะให้นิสิตเขียนโปรแกรมบนภาษาต่าง ๆ ด้วย, เพราะ
 - เพื่อความเข้าใจ
 - ความสนุก

รายละเอียดวิชา (1)

- เว็บ: <http://www.cpe.ku.ac.th/~jtf/204435>
 - จะ redirect ไปที่เว็บจริงที่อยู่บนวิกิอีกที
- เอกสาร
 - ส่วนใหญ่จะใช้สไลด์ของ John C. Mitchell
 - เอกสารคำบรรยายโดยผู้สอน (??)
 - เอกสารเพิ่มเติมบนอินเทอร์เน็ต
- การเตรียมตัวก่อนเข้าเรียน
 - นิสิตอาจต้องติดตั้งโปรแกรมมาก่อนเข้าเรียน
จะประกาศบนเว็บให้ติดตามด้วย

รายละเอียดวิชา (2)

- การให้คะแนน
 - การบ้าน 30%
 - การบ้านทั่วไป / โครงการ (ถ้ามี?)
 - สอบกลางภาค 30%
 - สอบปลายภาค 40%

ประวัติศาสตร์ (ช้าร่อย?)

- เทคโนโลยีที่เป็นไปได้ในภาษาโปรแกรม มีความสัมพันธ์อย่างยิ่งกับพัฒนาการด้านอื่น ๆ ของระบบคอมพิวเตอร์
 - สิ่งที่เป็นไปไม่ได้ในวันนี้ อาจเป็นสิ่งธรรมดาในอนาคต (ที่ไม่ไกล) ช่างหน้า?

ช้าร่อย?

- ในปี xxxx:
 - การเรียกตัวเอง (recursion) เป็นสิ่งไม่จำเป็น ทำให้คอมไพเลอร์ลำบาก และทำให้ประสิทธิภาพการทำงานต่ำ มีประโยชน์สำหรับการศึกษาเท่านั้น
 - ในปัจจุบันไม่มีภาษาสมัยใหม่ที่ไม่รองรับ recursion
 - ในบางภาษาสามารถจัดการกับ recursion บางรูปแบบได้รวดเร็วเท่า ๆ กับการเขียนด้วยการวนรอบธรรมดา (เช่นใน Scheme)

ช้าร่อย?

- ในปี xxxx:
 - การโปรแกรมเชิงวัตถุ เป็นแค่เรื่องของคนเท่านั้น
ไม่สามารถนำไปใช้ได้จริง
 - ปัจจุบันการโปรแกรมเชิงวัตถุได้ครอบครองโลกไปแล้ว

ช้าร่อย?

- ในปี xxxx:
 - การกำจัดขยะ (garbage collection) เป็นกระบวนการที่ไม่มีประสิทธิภาพ และไม่สามารถใช้งานได้จริง
 - มีอยู่ในทุกภาษาสมัยใหม่

ช้าร่อย?

- ในปี xxxx:
 - ภาษาโปรแกรมพลวัต (dynamic languages) หรือภาษาที่เป็นภาษาที่ต้องการแปล (interpreted languages) ทำงานช้า ไม่มีประสิทธิภาพ ไม่สามารถใช้งานที่รองรับความต้องการสูงได้
 - เว็บไซต์จำนวนมากพัฒนาด้วย php, ruby, perl

ช้าร่อย?

- ในปี xxxx:
 - ภาษาโปรแกรมเชิงฟังก์ชันเป็นเครื่องมือสำหรับทดลองในห้องวิจัยเท่านั้น
 - ดู F# ด้านหลัง

เรื่องเด่น

- แนวคิดต่าง ๆ ที่เราศึกษากำลังจะเป็นแนวโน้มใหม่
ของเทคโนโลยีการโปรแกรม (และบางอย่างกลายเป็น
สิ่งจำเป็นไปแล้ว)

เรื่องเด่น (1)

- ไมโครซอฟต์เพิ่มการรองรับภาษา F# ใน Visual Studio
 - F# พัฒนามาจาก Ocaml โดยเพิ่มการรองรับการใช้ไลบรารี .net
 - Ocaml เป็นสายพันธุ์หนึ่งของ ML ภาษาเชิงฟังก์ชันที่เราจะศึกษา

เรื่องเด่น (2)

- มีการพูดคุยเรื่องการทำให้ Java 7 รองรับ closure
 - Closure เป็นแนวคิดที่เริ่มพบในภาษาเชิงฟังก์ชัน

เรื่องเด่น (3)

- ภาษาแนวพลวัต (dynamic language) หรือที่มักโดนเรียกรวม ๆ ว่าภาษาสคริปต์ กลายมามีบทบาทสำคัญในการพัฒนาซอฟต์แวร์
 - เช่น Python, Ruby ภาษาเหล่านี้มีคุณลักษณะ น่าสนใจมากมายซึ่งทำให้การพัฒนาซอฟต์แวร์ด้วยภาษาเหล่านี้ทำได้ที่น่าสนใจ
 - Ruby on Rails และ Django (พัฒนาด้วย python) เป็น framework สำหรับพัฒนาโปรแกรมประยุกต์บนเว็บ ที่มีผู้ใช้เพิ่มขึ้นทุกวัน

การเตรียมตัวครั้งหน้า

- ดาวน์โหลดและติดตั้ง DrScheme มาก่อน