Today's Menu

Last Seminar:
Reuse Libraries
(Sreekanth led)

This Seminar:
Programming Languages
(Alejandra is leading)

Next Seminar:
Generative Methods
(Jon leads)

Little Languages -- what's the point?

Some your questions/comments

- How does "Little Languages" apply to software reuse?
- Can "Little Languages" handle complex problems nowadays?
- Are all the principles to create efficient "Little Languages" equally important in the philosophy of software reuse?
- What do "Little Languages" have to do with domain analysis/reuse?
- How is this paper related to what we've been studying in this course (software reuse)?
- I felt this paper was kind of off-topic ... didn't see where it could come into play in domain engineering.
- What's the connection? Are we wasting our time?

Little Languages (Cont'd)

Domain Engineering

Domain Engineering
Domain analysis
Domain design
Domain implementation

Application Engineering
(use the results from domain engineering to build software solutions)

Domain Engineering (Cont'd)

Domain Engineering
Domain analysis
Domain design
Domain implementation

1. Interactive (drag & drop)
2. Subroutine library
3. A "little language"

What do customers need? (what's sell?)
Why "Domain Engineering"?

Source Code
Linguistic Analysis
Code Generation
Object Code

Compiler

App. Code (SW src.)

Generator

Brian's domain
Nash's domain
Alejandra's domain
Are “Little Languages” Alive?

→ DSL (Domain-Specific Language)
  - Intent: each “little language” is appropriate for its problem domain
  - Examples: LaTeX, HTML, MATLAB, MAKE, SQL, Excel, etc.
  - FORTRAN (scientific computing), COBOL (business applications)
  - Bridge problem-domain concepts and solution-domain constructs
  - “linguistic insights”
  - How we talk about what we are doing influences how we work

→ “Programming Pearls” (very choice or precious) & Jon Bentley
  - B.S. in math (’74, Stanford), M.S. & Ph.D. (’76, UNC)
  - Students: James Gosling (father of Java), Joshua Block (CJA@Google)
  - Wrote the “Programming Pearls” column for CACM
  - “Programming Pearls”: gem; concise & focused essays; a celebration of design in the small; transcended the ordinary
  - A Google search “reviews Programming Pearls” returns ~60,000 records
  - “My biggest takeaway was Bentley’s authoritative yet humble tone. This stands in contrast to the high broadband and chest-thumping that defines so much technical communication in the current realm of blogs and tweets. These days it seems that every writer is a gladiator who’s foremost goal is to demonstrate technical superiority. It’s king of the cage for nerds. Bentley, … is certainly a king of the technical cage. Even so, he makes you feel like he’s a coworker discussing his current project during lunch.” — techguyinmidtown.com
  - Is it relevant to me? (e.g., Microsoft interview)

---

Look Ahead

<table>
<thead>
<tr>
<th>Date</th>
<th>Had Planned</th>
<th>Newly Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/15 M.</td>
<td>Jon</td>
<td>Jon</td>
</tr>
<tr>
<td>2/17 W.</td>
<td>Ken</td>
<td>Q&amp;As w.r.t. Project</td>
</tr>
<tr>
<td>2/22 M.</td>
<td>Chris</td>
<td>No Class (Project Due)</td>
</tr>
<tr>
<td>2/24 W.</td>
<td>Tan</td>
<td>Ken</td>
</tr>
<tr>
<td>3/1 M.</td>
<td>James</td>
<td>Chris</td>
</tr>
<tr>
<td>3/3 W.</td>
<td>-</td>
<td>Tan</td>
</tr>
<tr>
<td>3/8 M.</td>
<td>-</td>
<td>James</td>
</tr>
<tr>
<td>3/10 W.</td>
<td>Discuss Term Paper Topic</td>
<td>Discuss Term Paper Topic</td>
</tr>
</tbody>
</table>

Spring Break Spring Break Spring Break