Object-Oriented Programming

Conclusion

Michael L. Collard, Ph.D.

Department of Computer Science, The University of Akron

General Design Principles

- Consistency

- Hide as many details as possible, e.g., reduce the scope

- Only provide choice when necessary

- What does a "simple design" mean?

- Good defaults

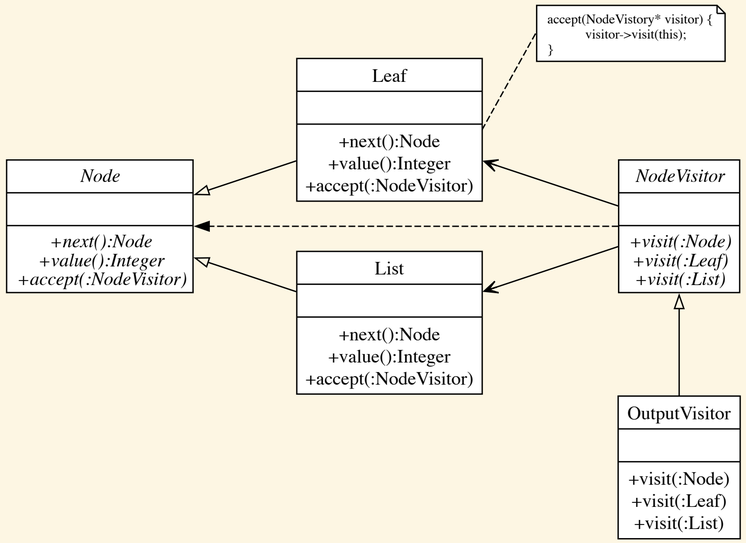
Importance of Naming

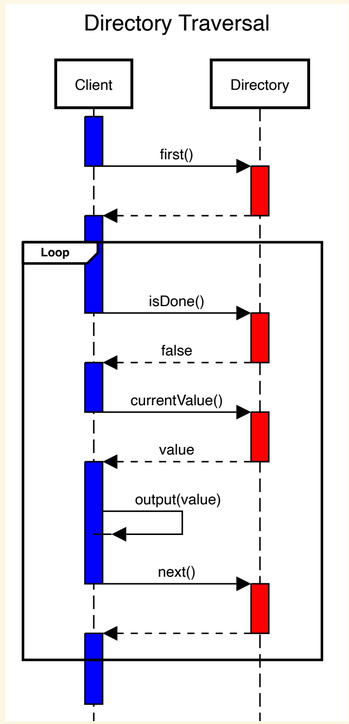
- Again, one of the most challenging parts of design

- Good names require extensive use of the system over a long time

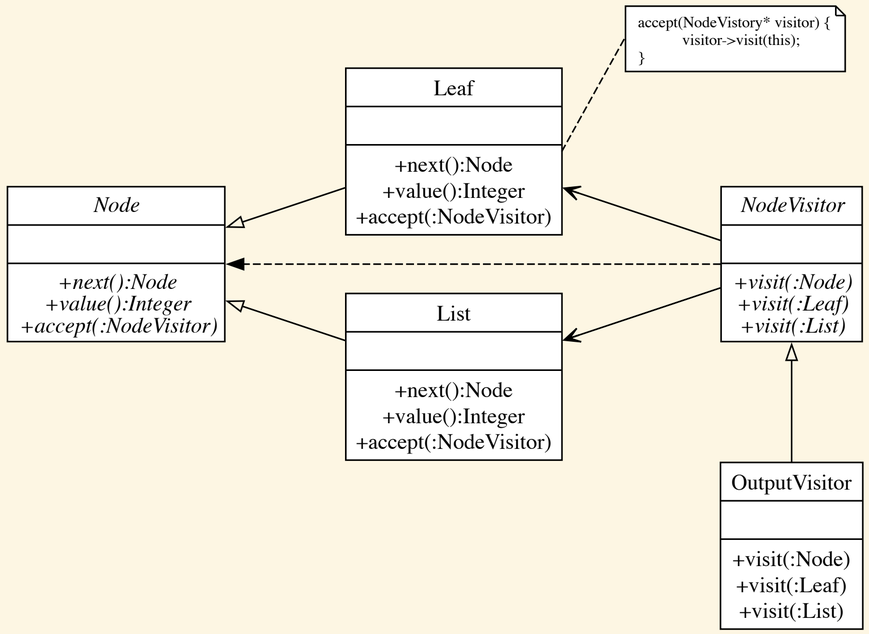
- For an API, they require lots of different examples

Modeling & Diagrams





Modeling & Diagrams



- It is very difficult to talk about modeling without the appropriate diagrams
- Used UML Class Diagrams and UML Sequence Diagrams, but there are many others
- Write code, then draw diagrams? Or draw diagrams, then code?
- *UmlAsSketch*

Design is about choices

- Quick fix (i.e., *technical debt*) vs. real solution
- Patch or wrapper vs. core change
- New features vs. old reliables
- Clarity vs. efficiency
- Code (low-level) vs. Models (high-level)

Change is Constant

- New language features to implement current design better

- New languages, including DSLs (Domain Specific Languages)

- New views on old design patterns and idioms

- New design patterns and idioms

Must be able to...

- Discuss design choices intelligently
- Be open to changing your thinking and your current design choices
- Keep up with new language features and new design ideas

Software & People

- Software is designed to help people and for people to use

- Software is designed and implemented by people

- Project skills are much more than just technical skills