Implementing an iterator for a binary search tree
Visitor Pattern Review
Visitor Pattern

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA, 1995.

• Intent:
  - “Represent an operation to be performed on the elements of an object structure. Visitor lets you define a new operation without changing the classes of the elements on which it operates.”

• When to Use:
  - Many classes of objects
  - Distinct and unrelated operations to perform on object
  - Object structures rarely change but add new operations often

• Benefits:
  - Adding new operations is easy
  - Gathers related operations and separate unrelated ones
  - Accumulating state
Parameterized Types/Generics
“Generic types and methods provide a way to strengthen type checking at compile-time while at the same time making programs more expressive, reusable and readable. The ability to have generic types and methods is also known as parametric polymorphism.”
A little syntax...

- class name<T1, T2, ..., Tn>
- method-modifiers <T1, ..., Tn>
  returntype m(formal-list)

- Wildcards
  - <?>
  - <? extends tb>
  - <? super tb>
Generic types that we have used this semester

- `ArrayList<E>`
- `Comparator<T>`
- `Iterable<T>`
- `Iterator<E>`
Generics
[Bloch]

• Item 23: Don’t use raw types in new code.
• Item 24: Eliminate unchecked warnings
• Item 25: Prefer lists to arrays
• Item 26: Favor generic types
• Item 27: Favor generic methods
• Item 28: Use bounded wildcards to increase API flexibility