Readings for this week...

• Liskov Chapter 8
• Liskov Chapter 7
• Bloch Chapter 5
Assignment 6
Red-Black Tree Review
Red-Black Trees

Red-black trees are binary search trees in which each node has a color (either red or black) and the following balancing invariants are preserved by all operations:

1. No red node has a red child.

2. Every path from the root to an empty tree/node contains the same number of black nodes.
balance $B (T R (T R a x b) y c) z d$
$= T R (T B a x b) y (T B c z d)$

balance $B (T R a x (T R b y c)) z d$
$= T R (T B a x b) y (T B c z d)$

balance $B a x (T R (T R b y c) z d)$
$= T R (T B a x b) y (T B c z d)$

balance $B a x (T R b y (T R c z d))$
$= T R (T B a x b) y (T B c z d)$

balance color $a x b = T$ color $a x b$

from Okasaki
ICE
StringByLex Comparator

1. state
2. university
3. wolfpack
4. north
5. carolina
6. ncsu
7. computer
8. science
9. sciences

from Okasaki
Visitor Pattern
Visitor Pattern

• “The visitor pattern is a general design pattern that allows you to separate data from functionality in an object-oriented style.”

• “[I]t requires you to implement one method which accepts what we call a ‘visitor’ that is then exposed to a view of the data.”
Visitor Pattern

[Gamma et al.]

“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 Visitor Pattern

[Gamma et al.]

• Many classes of objects
• Distinct and unrelated operations to perform on object
• Object structures rarely change but add new operations often
Benefits of Using Visitor Pattern

[Gamma et al.]

• 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.”
public class Box {
    private Object object;

    public void set(Object object) {
        this.object = object;
    }

    public Object get() {
        return object;
    }
}

Code from: http://docs.oracle.com/javase/tutorial/java/generics/types.html
/**
 * Generic version of the Box class.
 * @param <T> the type of the value being boxed
 */

class Box<T> {
    // T stands for "Type"
    private T t;

    public void set(T t) { this.t = t; }
    public T get() { return t; }
}
Generics Terminology

[Bloch]

• type parameters
• generic class or interface
• parameterized types
• actual type parameters
• raw type
• unbounded wildcard types
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