Thursday, February 15

Warmup
What is a class invariant? How do you use it? How is it enforced?

The class invariant is a set of rules about what should be true about an instance of the class, as represented by its instance variables, outside any methods of the class. It is a checklist of what is true about the instance variables when a method starts and what should be true when a method finishes. The class’s implementor maintains it; there is no automatic checking. The text refers to this (on p.126) as the invariant of the ADT.

Statistician
Due Monday
most of you well on your way to completing it after lab Tuesday
JUnit runs test cases of input and expected output - StatTest

Chapter 3 - Collection Classes
recurring example of Bag
understand the abstract class
represent it in many ways
user of Bag does not need to know how it is built
information hiding (hidden from user and not from developer)

items
goal of building container class independent of type of items
use one explicit item type now
but try to think about the container and not the types of items

what is a bag?
starts empty
no order
duplicates allowed

LetterArrayBag
LetterArrayBag plasticLetters = new LetterArrayBag(20);
LetterArrayBag marks, helens;   // variables not connected to bag
marks = new LetterArrayBag( );   // new bag with default size
marks.add('M');
helens = marks.clone( );   // creates new one and duplicates contents
LetterArrayBag davids = marks;   // same bag with two variables
davids.add('S');   // adds S to bag that is referred to by marks and davids
Extended Example - Int ArrayBag

HTML is Javadoc specification
WHAT not the HOW

    // Invariant of the IntArrayBag class:
    // 1. The number of elements in the bag is in the instance variable
    //    manyItems, which is no more than data.length.
    // 2. For an empty bag, we do not care what is stored in any of data;
    //    for a non-empty bag, the elements in the bag are stored in data[0]
    //    through data[manyItems-1], and we don’t care what’s in the
    //    rest of data.
    private int[ ] data;
    private int manyItems;

Assignment - Sequence with Array

    // Invariant of the DoubleArraySeq class:
    // 1. manyItems is the number of elements in the sequence.
    // 2. For an empty sequence (with no elements), we do not care what is
    //    stored in any of data; for a non-empty sequence, the elements of the
    //    sequence are stored in order in data[0] through data[manyItems-1],
    //    and we don’t care what’s in the rest of data.
    // 3. If there is a current element, then it is in data[currentIndex];
    //    if there is no current element, then currentIndex equals manyItems.
    private double[ ] data;
    private int manyItems;
    private int currentIndex;