Java By Abstraction - Companion Notes
Topic 7 – Strings

CSE 1720, Winter 2010, Version 1.0, Prepared by: M. Baljko

corresponding to Chapter 6 of JBA

Section 6.1; PT 6.1-6.2; Ex 6.1

- note the statement that Strings are allowed to masquerade as primitive types in Java – this statement is important, since we will see that Strings are non-primitive types, but in many instances have the appearance of being a primitive type.
- 6.1.1 – talks about String object creation, with illustrations using memory diagrams
- PT 6.1 talks about the empty string vs null (aka "the null string"); the empty string is represented by the literal `""` (i.e., a set of double quotes with nothing in between them)
- 6.1.2 – reinforcement of what we've already seen in lab and lecture, a good review
- PT 6.2 important point

Section 6.2; JD 6.1; PT 6.3

- we've already used several of these string methods; this section provides a review and the introduction of other methods.
- Notice that sec 6.2.3 is entitled "Transformers" (as opposed to "Mutators"; the duo of "Accessors" and "Mutators" as categories of methods were discussed in sec 4.3.1) – it should be clear that one cannot mutate or change a string once it has been created; at best, one can create a modified version of a string and then use that instead.
- sec 6.2.5 is important, since it described how to shift from one type to another. Also note PT 6.3, which applies to this functionality.
- JD 6.1 makes mention of boxing and unboxing – this point has to do with the automatic transformation of a value between the primitive and non-primitive representations (e.g., the value of 6 can be represented as an `int` or as an instance of an `Integer`). Refer back to p. 123 and the description of a *wrapper* class.

Section 6.3

- this section describes several different applications
- 6.3.1 concerns counting the number of occurrences of a particular character within a string. This example does not make use of delegation, but it easily could. Imagine a method in a utility class with the following signature:
  `countOccurrences(String theString, char c)`
  As an exercise, consider the example app given in fig 6.3. Delegate the corresponding functionality into a static method (use the Mortgage and Pig Latin applications we developed as templates)
- 6.3.2, 6.3.3, 6.3.4 all concerns substitutions of characters and of strings (two versions: one assuming new and old string are old equal length and the other version with no such assumption). Again, it would be a good ideal to use delegation

Section 6.4

- 6.4.1 StringBuffer is important; think of it as the mutable version of String
- 6.4.2 is an advanced topic; regular expressions are an important concept not only for Java but for any sort of pattern matching. We will do several examples in lecture/lab. Read this subsection in order to prepare.