Stephen Pillsbury Exam 2 CSC360 Notes

Chapters 16-18

Ummm, glhf?

Chapter 19

Example: public interface Comparable<T>{

public int compareTo(T o)

}

Generic methods allow you to specify which types of objects a method is allowed to handle, flexibly.

<T> represents a formal generic type, which is replaced at runtime by an actual concrete type. Replacing a generic type is called generic instantiation.

Generic types must be reference types, as opposed to primitive types like int. However, primitive types can be added via autoboxing: intList.add(5);

A wildcard generic type has three forms: ?, ? extends T, and ? Super T, where T is a generic type. The first form, is called an unbounded wildcard, is the same as ? Extends Object. (It covers pretty much everything.) The second form, ? Extends T, is called a bounded wildcard, and represents T or a subtype of T. The third form, ? Super T, is called a lower-bound wildcard, and represents T or a supertype or T.

Generics are implemented using type erasure, meaning generic type information is used to compile the code but erased afterwards, so the generic information is unavailable at runtime.

Chapter 20

Stack class extends collection framework, is last in, first out. Can pop, peek, push, check for empty, and search for an object.

Arraylist is a dynamically created array implementation of the list interface, and expands by copying itself into a larger memory location. Objects inside are referred to by their index. It doesn’t automatically shrink.

Linked lists extend the list interface, and have operations to add elements at the beginning and end of the list. Can also add objects at given indices with add(). One can iterate through a linked list with a listiterator object.

The comparator interface can be used to compare objects of classes that don’t implement Comparable. Comparator uses generics via “ public int compare(T element1, T element2){}”, which returns – if e1 < e2, + if e2 < e1, and 0 if equal.

You can then compare two attributes of the parameter objects and return accordingly.

Comparatively, objects implementing the comparable interface can just call ObjName.compare(obj2).