Ch 9 Class Activity

ITEC 3860 - Heinz

Jeremy Cooley

1. Consider the List interface in the java.util package for ordered collections of objects. Write preconditions and/or post conditions for the following operations:
   1. void add(Object e) adds an object at the end of the list.  
      context List :: add(e)   
       Pre: input = element  
       Post: contains(e) && size() = pre.size() + 1
   2. void remove(Object e) removes the first occurrence of an object from the the list.  
      context List :: remove(e)   
       Pre: input = element && contains(e)  
       Post: pre.contains(e) && (@pre.contains(e) implies size() = pre.size() – 1) && (not pre.contains(e) implies size() = pre.size())
   3. Object get(int index) returns the object located at index, 0 being the index of the first object in the list.  
      context List :: get(index)   
       Pre: 0 <= index <= list.size() – 1  
       Post: element @ list[ index ]
2. Consider the Set interface in the java.util package. Write preconditions and/or post conditions for the following operations:
   1. void add(Object e) adds an object to the set. If the object is already in the set, does nothing.  
      context Set :: add(e)   
       Post: contains(e) && (@pre.contains(e) implies size() = pre.size() ) && (not @pre.contains(e) implies size() = pre.size() + 1)
   2. void remove(Object e) removes an object from the set  
      context Set :: remove(e)  
       Post: not contains(e) && (@pre.contains(e) implies size() = pre.size() – 1 ) && (not @pre.contains(e) implies size() = pre.size() )
   3. boolean contains(Object e) returns true if the object is contained in the set.  
      context Set :: contains(e)   
       Post: elements->includes(e)