2. Sorting Algorithms :: Insertion Sort

Here is a Java implementation of insertion sort that sorts an ArrayList<T> list (where T can be any class which implements the *Comparable*<T> interface).

```
//*************************************************************************************
// CLASS: InsertionSorter (InsertionSorter.java)
//*************************************************************************************
import java.util.ArrayList;

public class InsertionSorter {

    public static final int SORT_ASCENDING  = 0;
    public static final int SORT_DESCENDING = 1;

    /**
     * Sorts pList into ascending (pOrder = SORT_ASCENDING) or descending (pOrder =
     * SORT_DESCENDING) order using the insertion sort algorithm.
     */
    public static void insertionSort(ArrayList<T> pList, int pOrder) {
        for (int i = 1; i < pList.size(); ++i) {
            for (int j = i; keepMoving(pList, j, pOrder); --j) {
                swap(pList, j, j - 1);
            }
        }
    }
```

2. Sorting Algorithms :: Insertion Sort (continued)

```
    /**
     * Returns true if we need to continue moving the element at pIndex until it reaches its
     * proper location. If pIndex is less than 1, then the element that we were moving has been
     * moved to index 0 of list so we are done. Otherwise, we compare the element at pIndex to
     * the element that precedes it in the list -- at pIndex - 1 -- and if they are out of order
     * we swap them.
     */
    private static boolean keepMoving(ArrayList<T> pList, int pIndex, int pOrder) {
        if (pIndex < 1) return false;
        if (pOrder == SORT_ASCENDING) {
            return pList.get(pIndex).compareTo(pList.get(pIndex - 1)) < 0;
        } else {
            return pList.get(pIndex).compareTo(pList.get(pIndex - 1)) > 0;
        }
    }

    /**
     *  Swaps the elements in pList at pIndex1 and pIndex2.
     */
   private static void swap(ArrayList<T> pList, int pIndex1, int pIndex2) {
        T temp = pList.get(pIndex1);
        pList.set(pIndex1, pList.get(pIndex2));
        pList.set(pIndex2, temp);
    }
}
```