

AP COMPUTER SCIENCE A – HOMEWORK #9

The goal of this assignment is to write a program which sorts the list of presidents (contained in the file `presidents.txt`, which lists the presidents in the order that they served) into alphabetical order according to their last names, then first names, then middle names (or initials). The final program will consist of several classes (and an interface), as will be described below.

1) Add two methods to file `ListStub.java` to make a class `List<T>`, viz.,

```
a)  /*
    *   Remove the element at position index from the list. Elements originally at
    *   indices index+1 to length-1 (inclusive) are moved up one spot.
    */
    public void remove ( int index ) {
        /* add necessary code */
    }
```

Be sure to call `checkIndexBounds(index)` at the start of the method, and then to decrement `length` at the end.

```
b)  /*
    *   Insert element e into the list at position index. Elements originally at
    *   indices index to length-1 (inclusive) are moved down one spot.
    */
    public void insert ( T e, int index ) {
        /* add necessary code */
    }
```

If `index == length`, then just call `add(e)` and return. Otherwise, do

```
checkIndexBounds(index);
++length;
if ( length == numBlocks*BLOCK_SIZE ) grow();
```

and then move the necessary entries down one spot and insert `e` at the proper location.

2) Write a class `Name` implements `Sortable`, where `Sortable` is the interface

```
public interface Sortable {
    boolean lessThan( Sortable sortable );
}
```

- a) The constructor should have the signature `public Name (String s)`, where `s` will be one of the presidents' names, e.g., James K Polk or Zachary Taylor. In the constructor, you should construct a comparable string, such as `PolkJamesK` or `TaylorZachary`, and make both the original name string `s` and the so-constructed comparable string private fields of the class.
- b) Your class should contain a `public String toString()` method which just returns the president's original name (as it was read in), e.g., James K Polk or Zachary Taylor.
- c) Your class should have a public accessor method for the comparable name string, which accessor you will have to use in the `public boolean lessThan (Sortable sortable)` method.
- d) When implementing `public boolean lessThan (Sortable sortable)`, you will have to cast `sortable` to a `Name`, and then use the `int compareTo (String anotherString)` method of class `String` to compare two of the comparable name strings.

AP COMPUTER SCIENCE A – HOMEWORK #9

- 3) Modify the file `InsertionSortStub.java` to make a class `InsertionSort` containing just one method, *i.e.*, `public static <T extends Sortable> void insertionSort (List<T> list)`. To write this method, modify the original `public long insertionSort ()` method by

- a) removing the timer,
- b) declaring `bot` as a `T`,
- c) making `int [] list` a `List<T>`,
- d) changing

```
if ( list[a] > bot ) topI=a;

to

if ( bot.lessThan(list.get(a)) ) topI=a;
```

- e) and finally, by changing

```
for ( int m=i; m>topI; --m ) list[m]=list[m-1];
list[topI]=bot;

to

list.remove(i);
list.insert(bot,topI);
```

- 4) Finally, write a class `SortPresidents.java` containing a main method which ties all of the above code together. The program `SortPresidents` should

- a) read the file `presidents.txt` into a `List<Name>` object, *e.g.*, `List<Name> nameList`,
- b) perform the sort via `InsertionSort.<Name>insertionSort(nameList)`;
- c) and finally, print out to a file `sorted_presidents.txt`, using a `FileOutput fo`, the sorted list using a for-each loop, *i.e.*,

```
for ( Name n : nameList ) {
    fo.println(n);
}
```

- 5) Submit your files `List.java`, `Name.java`, `InsertionSort.java` and `SortPresidents.java`.