【DS】Day6

Tags	
Date	@May 28, 2022
Summary	Sorting Client and Comparable Interface

【Week2】Sorting

2.3 Sorting Introduction

Goal: Sort any type of data

Ex 1: Sort random real numbers in ascending order

Test Client

Sort Double

```
public class Experiment {
  public static void main(String[] args) {
    int N = Integer.parseInt(args[0]);
    Double[] arr = new Double[N];
    for (int i = 0; i < N; ++i) {
      arr[i] = StdRandom.uniform();
    }
    Insertion.sort(arr);
    for (int i = 0; i < N; ++i) {
      StdOut.println(arr[i]);
    }
  }
}
```

Sort String

```
public class StringSorter {
  public static void main(String[] args) {
    String[] a = In.readStrings(args[0]);
    Insertion.sort(a);
    for (int i = 0; i < a.length; ++i)
      StdOut.println(a[i];
```

```
  }
}
```

Comparable Interface(built in to Java)

```
public interface Comparable<Item> {
  public int compareTo(Item that);
}
```

Object Implementation:

```
public class File implements Comparable<File> {
  public int compareTo(File b) {
    ...
    // This object is less than the passed-in object
    return -1;
    ...
    // This object is greater than the passed-in object
    return +1;
    ...
    // Two objects are equal
    return 0;
  }
}
```

Sort Implementation:

```
public static void sort(Comparable[] a) {
  int N = a.length;
  for (int i = 0; i < N; ++i) {
    for (int j = i; j > 0; --j) {
      // If the jth object is less than the j-1th object
      if(a[j].compareTo(a[j - 1]) < 0)
        exch(a, j, j - 1);
      else
        break;
    }
  }
}
```

Comparable API

Implement `compareTo()` so that `v.compareTo(w)`

- Is a total order
- Returns a negative integer, zero, or positive integer

  if v is less than, equal to, or greater than w, respectively.
- Throw an exception if both or either object is null

Helper functions: Refer to data through compares and exchanges:

```
private static boolean less(Comparable v, Comparable w) {
  return v.compareTo(w) < 0;
}
```

Exchange: Swap item in array `a[]` at index i with the one at index j

```
private static void exch(Comparable[] a, int i, int j) {
  Comparable swap = a[i];
  a[i] = a[j];
  a[j] = swap;
}
```

Test if an array is sorted:

```
private static boolean isSorted(Comparable[] a) {
  for (int i = 1; i < a.length; ++i) {
    if (less(a[i], a[i - 1))
      return false;
  }
  return true;
}
```