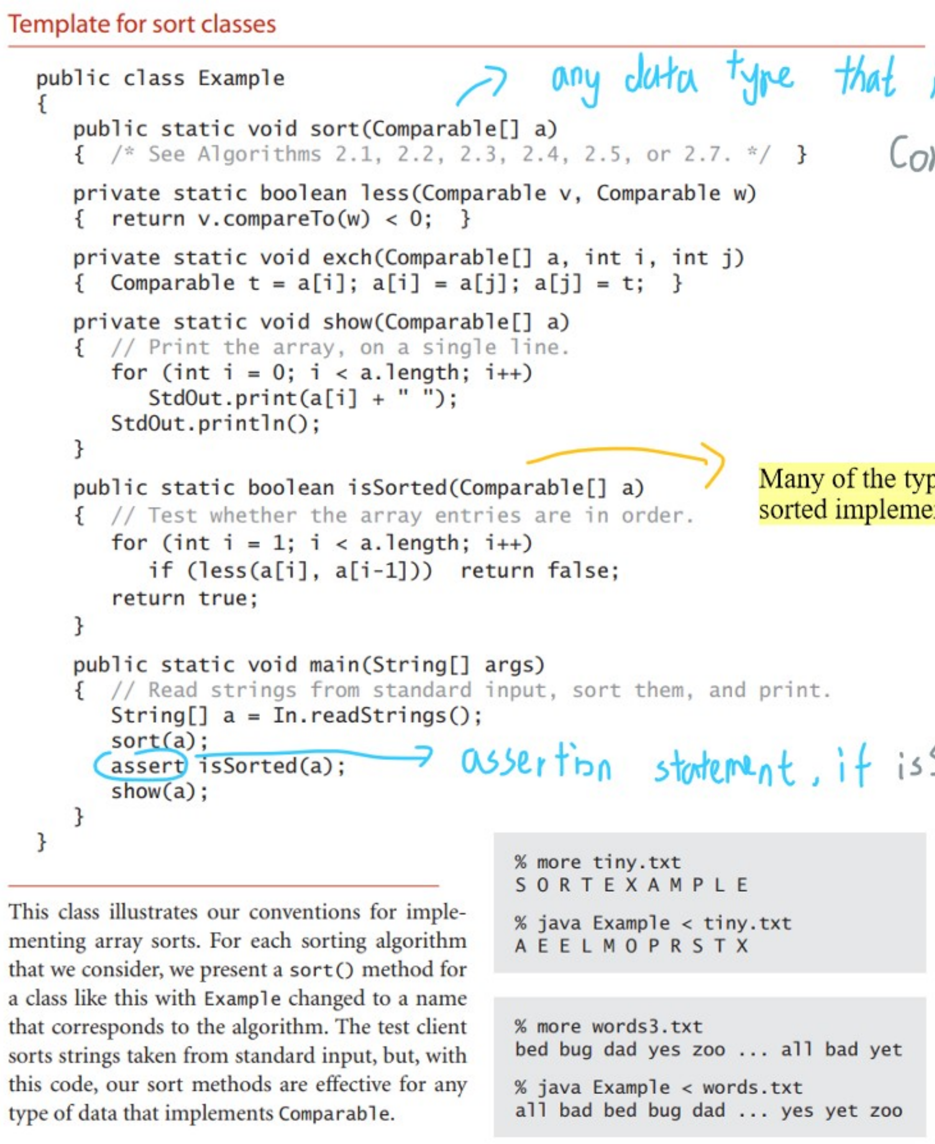
Elementary Sorts Intro

2022年10月20日 13:22



Template for sort classes

```
public class Example
{
   public static void sort(Comparable[] a)
   {  /* See Algorithms 2.1, 2.2, 2.3, 2.4, 2.5, or 2.7. */  }

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

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

   private static void show(Comparable[] a)
   {  // Print the array, on a single line.
      for (int i = 0; i < a.length; i++)
         StdOut.print(a[i] + " ");
      StdOut.println();
   }

   public static boolean isSorted(Comparable[] a)
   {  // Test whether the array entries are in order.
      for (int i = 1; i < a.length; i++)
         if (less(a[i], a[i-1]))  return false;
      return true;
   }

   public static void main(String[] args)
   {  // Read strings from standard input, sort them, and print.
      String[] a = In.readStrings();
      sort(a);
      assert isSorted(a);
      show(a);
   }
}
```

any data type that implements the Comparable interface

Many of the types of data that need to be sorted implement Comparable

assertion statement, if isSorted(a) is not true, the program will stop running. assertion is great for debugging errors.

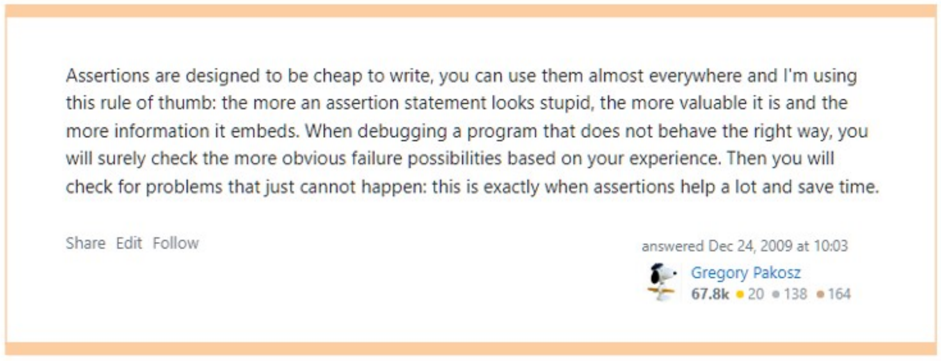
This class illustrates our conventions for implementing array sorts. For each sorting algorithm that we consider, we present a sort() method for a class like this with Example changed to a name that corresponds to the algorithm. The test client sorts strings taken from standard input, but, with this code, our sort methods are effective for any type of data that implements Comparable.

```
% more tiny.txt
S O R T E X A M P L E

% java Example < tiny.txt
A E E L M O P R S T X
```

```
% more words3.txt
bed bug dad yes zoo ... all bad yet

% java Example < words.txt
all bad bed bug dad ... yes yet zoo
```



Assertions are designed to be cheap to write, you can use them almost everywhere and I'm using this rule of thumb: the more an assertion statement looks stupid, the more valuable it is and the more information it embeds. When debugging a program that does not behave the right way, you will surely check the more obvious failure possibilities based on your experience. Then you will check for problems that just cannot happen: this is exactly when assertions help a lot and save time.

Share Edit Follow

answered Dec 24, 2009 at 10:03

Gregory Pakosz

67.8k ● 20 ● 138 ● 164

An example of an abstract that implement Comparable interface

```
public class Date implements Comparable<Date>
{
   private final int day;
   private final int month;
   private final int year;

   public Date(int d, int m, int y)
   {  day = d; month = m; year = y; }

   public int day()   {  return day;    }
   public int month() {  return month;  }
   public int year()  {  return year;    }

   public int compareTo(Date that)   required for Comparable implementation
   {
      if (this.year  > that.year ) return +1;
      if (this.year  < that.year ) return -1;
      if (this.month > that.month) return +1;
      if (this.month < that.month) return -1;
      if (this.day   > that.day  ) return +1;
      if (this.day   < that.day  ) return -1;
      return 0;
   }

   public String toString()
   { return month + "/" + day + "/" + year; }
}
```