CZ2002 LAB 4

OBJ ORIENTED DES & PROG

SS5 GROUP 6

Zou Zeren U2022422H

Contents

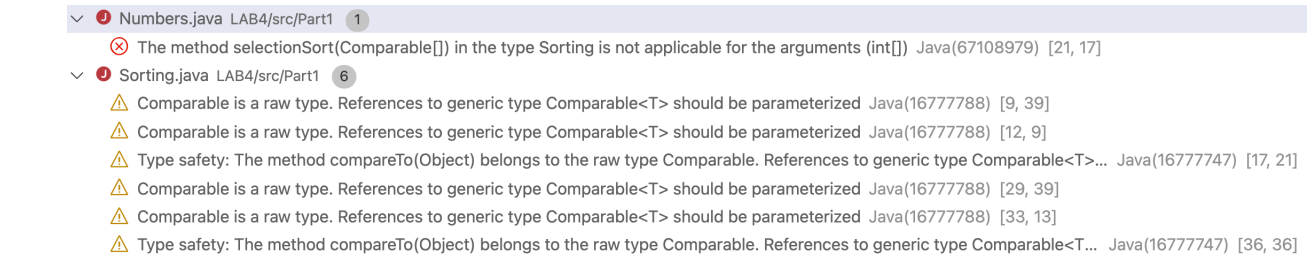
1 Numbers.java Issue

The file Numbers.java (in attachment) reads in an array of integers, invokes the selection sort algorithm to sort them, and then prints the sorted array. Save Sorting.java and Numbers.java to your directory. Numbers.java won't compile in its current form. Study it to see if you can figure out why.

Numbers.java have to use functions from the Sorting.java.
Int is a primitive type, and therefore cannot implement any interface. That's why int[] cannot be passed to a method that expects Comparable[]. To overcome this error, change intList to be an array of Integer (i.e Integer[]), since Integer implements Comparable.

2 Numbers.java Errors

Try to compile Numbers.java and see what the error message is. The problem involves the difference between primitive data and objects. Change the program so it will work correctly (note: you don't need to make many changes - the autoboxing feature of Java 1.5 (or higher) will take care of most conversions from int to Integer). You are to do research in the internet and understand better autoboxing.




```
Run | Debug
public static void main (String[] args)
{
    int[] intList;
    int size;
    Scanner scan = new Scanner(System.in);
    System.out.print ("\nHow many integers do you want to sort? ");
    size = scan.nextInt();
    intList = new int[size];
    System.out.println ("\nEnter the numbers...");
    for (int i = 0; i < size; i++)
        intList[i] = scan.nextInt();
    Sorting.selectionSort(intList);
    System.out.println ("\nYour numbers in sorted order...");
    for (int i = 0; i < size; i++)
        System.out.print(intList[i] + " ");
    System.out.println ();
}
}
```

```
Run | Debug
public static void main (String[] args)
{
    Integer[] intList;
    int size;
    Scanner scan = new Scanner(System.in);
    System.out.print ("\nHow many integers do you want to sort? ");
    size = scan.nextInt();
    intList = new Integer[size];
    System.out.println ("\nEnter the numbers...");
    for (int i = 0; i < size; i++)
        intList[i] = scan.nextInt();
    Sorting.selectionSort(intList);
    System.out.println ("\nYour numbers in sorted order...");
    for (int i = 0; i < size; i++)
        System.out.print(intList[i] + " ");
    System.out.println ();
}
}
```

```
How many integers do you want to sort? 4

Enter the numbers...
3
4
2
1

Your numbers in sorted order...
1 2 3 4
anthonyzeren@Zous-MacBook-Pro CZ2002-OODP-NTU %
```

3 String.java

Write a program Strings.java, similar to Numbers.java, that reads in an array of String objects and sorts them. You may just copy and edit Numbers.java.

```java
package Part1;

import java.util.Scanner;
public class Numbers
{
    // -----------------------------------------
// Reads in an array of integers, sorts them,
// then prints them in sorted order.
// -----------------------------------------
    Run | Debug
    public static void main (String[] args)
    {
        Integer[] intList;
        int size;
        Scanner scan = new Scanner(System.in);
        System.out.print ("\nHow many integers do you want to sort? ");
        size = scan.nextInt();
        intList = new Integer[size];
        System.out.println ("\nEnter the numbers...");
        for (int i = 0; i < size; i++)
            intList[i] = scan.nextInt();
        Sorting.selectionSort(intList);
        System.out.println ("\nYour numbers in sorted order...");
        for (int i = 0; i < size; i++)
            System.out.print(intList[i] + " ");
        System.out.println ();
        scan.close();
    }
}
```

```
How many string do you want to sort? 4

Enter the strings...
apple
pear
water melon
banana

Your strings in sorted order...
 apple banana pear water melon
```

4 Decending Insertion Sort

Modify the insertionSort algorithm so that it sorts in descending order rather than ascending order. Change Numbers.java and Strings.java to call insertionSort rather than selectionSort. Run both to make sure the sorting is correct.

Decending Insertion Sort

```java
//-----------------------------------------------------------------
// Sorts the specified array of objects using the insertion
// sort algorithm.
//-----------------------------------------------------------------
public static void insertionSort (Comparable[] list)
{
    for (int index = 1; index < list.length; index++)
    {
        Comparable key = list[index];
        int position = index;
        // Shift larger values to the right
        //while (position > 0 && key.compareTo(list[position-1]) < 0)
        while (position > 0 && key.compareTo(list[position-1]) > 0)
        {
            list[position] = list[position-1];
            position--;
        }
        list[position] = key;
    }
}
```

Tested on Numbers

```java
package part1;

import java.util.Scanner;
public class Numbers
{
    // -----------------------------------------
// Reads in an array of integers, sorts them,
// then prints them in sorted order.
// -----------------------------------------
    Run | Debug
    public static void main (String[] args)
    {
        Integer[] intList;
        int size;
        Scanner scan = new Scanner(System.in);
        System.out.print ("\nHow many integers do you want to sort? ");
        size = scan.nextInt();
        intList = new Integer[size];
        System.out.println ("\nEnter the numbers...");
        for (int i = 0; i < size; i++)
            intList[i] = scan.nextInt();
       // Sorting.selectionSort(intList);
        Sorting.insertionSort(intList);
        System.out.println ("\nYour numbers in sorted order...");
        for (int i = 0; i < size; i++)
            System.out.print(intList[i] + " ");
        System.out.println ();
        scan.close();
    }
}
```

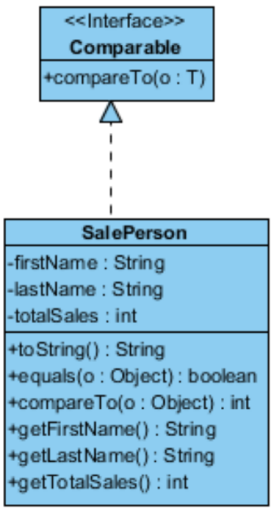
```
How many integers do you want to sort? 3

Enter the numbers...
2
1
3

Your numbers in sorted order...
3 2 1
```

Tested on Strings

```java
package part1;

import java.util.Scanner;
public class Strings {
    // ------------------------------------------
// Reads in an array of integers, sorts them,
// then prints them in sorted order.
// ------------------------------------------
Run | Debug
public static void main (String[] args)
{
    String[] strList;
    int size;
    Scanner scan = new Scanner(System.in);
    System.out.print ("\nHow many string do you want to sort? ");
    size = scan.nextInt()+1;
    strList = new String[size];
    System.out.println ("\nEnter the strings...");
    for (int i = 0; i < size; i++)
        strList[i] = scan.nextLine();
    //Sorting.selectionSort(strList);
    Sorting.insertionSort(strList);
    System.out.println ("\nYour strings in sorted order...");
    for (int i = 0; i < size; i++)
        System.out.print(strList[i] + " ");
    System.out.println ();
    scan.close();
}
}
```

```
How many string do you want to sort? 3

Enter the strings...
apple
banana
cherry

Your strings in sorted order...
cherry banana apple
```

5 SalesPerson Class



5. The class diagram on the right defines the SalePerson class that represents a sale person. The sale person has a first name, last name, and a total number of sales (an int).

- The *toString* method will return the name of the sale person and total sales in the formal :
  <lastName> , <firstName> : <totalSales>
- The *equals* method will check whether the first and last names of Object are the same as the current sale person.
- The *compareTo* method make the comparison based on total sales; that is, return a negative number if the executing object has total sales less than the other object and return a positive number if the sales are greater. ***Use the name of the sales person's last name to break a tie (in ascending alphabetical order).***

- ***Create and Write the SalePerson class***