```java
import java.util.ArrayList;
import java.util.Comparator;
import java.util.Random;

/**
 * File: QuickSort.java
 *
 * Implementation von Quicksort. Verwendet generische ArrayList für Eingabedaten
 * und generischen Comparator um Datenelemente zu vergleichen.
 */
public class QuickSort {

/**
 * Sortiert den array zwischen den Indizes left und right (mit Grenzen).
 */
public static <T> void quickSort(ArrayList<T> array, int left, int right, Comparator<T> comp) {
        if (right>left) {  // Abbruchbedingung der Rekursion
                T temp;        // temporaere Hilfsvariable zum swappen

                // *** 0. Pivot zufällig austauschen
                Random generator = new Random();
                int rand = generator.nextInt(right-left) + left;

                T t = array.get(rand);
                array.remove(rand);
                array.add(right,t);

                // *** 1. Pivotelement selektieren:
                T pivot = array.get(right);

                // *** 2. Aufteilung in Subsequenzen durchfuehren:
                int l = left-1;
                int r = right;
                do {
                        do l++; while (comp.compare(array.get(l),pivot)<0);
                        do r--; while (r>l && comp.compare(array.get(r),pivot)>0);

                        // swap(array, l, r):
                        temp = array.get(l);
                        array.set(l, array.get(r));
                        array.set(r, temp);
                } while (r>l);

                array.set(r, array.get(l));   // Korrektur: einmal zuviel getauscht

                // Pivotelement in sortierte Position bringen:
                array.set(l, pivot);
                array.set(right, temp);

                // *** 3. Rekursiv die beiden Subarrays sortieren:
                quickSort(array, left, l-1, comp);
                quickSort(array, l+1, right, comp);
        }
        }
}
```