Create a simple generics class with type parameters for sorting values of different types.

CODE:

import java.util.Arrays;

import java.util.Scanner;

class GenericSort<T extends Comparable<T>> {

private T[] array;

public GenericSort(T[] array) {

this.array = array;

}

public void sort() {

Arrays.sort(array);

}

public void printSortedArray() {

System.out.println("Sorted Array:");

for (T value : array) {

System.out.print(value + " ");

}

System.out.println();

}

}

public class GenericsSortingExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements: ");

int numElements = scanner.nextInt();

Integer[] intArray = new Integer[numElements];

Double[] doubleArray = new Double[numElements];

Character[] charArray = new Character[numElements];

System.out.println("Enter " + numElements + " integers:");

for (int i = 0; i < numElements; i++) {

intArray[i] = scanner.nextInt();

}

System.out.println("Enter " + numElements + " doubles:");

for (int i = 0; i < numElements; i++) {

doubleArray[i] = scanner.nextDouble();

}

System.out.println("Enter " + numElements + " characters:");

for (int i = 0; i < numElements; i++) {

charArray[i] = scanner.next().charAt(0);

}

scanner.close();

GenericSort<Integer> intSorter = new GenericSort<>(intArray);

intSorter.sort();

intSorter.printSortedArray();

GenericSort<Double> doubleSorter = new GenericSort<>(doubleArray);

doubleSorter.sort();

doubleSorter.printSortedArray();

GenericSort<Character> charSorter = new GenericSort<>(charArray);

charSorter.sort();

charSorter.printSortedArray();

}

}

OUTPUT:

C:\javap>javac GenericsSortingExample.java

C:\javap>java GenericsSortingExample

Enter the number of elements: 5

Enter 5 integers:

7 5 3 9 5 1

Enter 5 doubles:

1 2 3 6 8

Enter 5 characters:

k wy s i h

Sorted Array:

3 5 5 7 9

Sorted Array:

1.0 1.0 2.0 3.0 6.0

Sorted Array:

8 i k s w

