**Project**

**Algorithm-Benchmarking**

import java.util.Arrays;

import java.util.Collections;

import java.util.Scanner;

import javax.security.auth.DestroyFailedException;

public class Test {

public static void main(String[] args) {

long start,stop,total;

Scanner obj= new Scanner(System.in);

System.out.println("enter size of array");

int size = obj.nextInt();

System.out.println("choose comliplexcity \n\n 1. Best case \n\n 2.Avg case \n\n 3.Worst case \n\n ");

int n = obj.nextInt();

int [] array =new int [size];

int [] bubblearray =new int [size];

int [] selectionarray =new int [size];

switch (n){

case 1:{

array=Calc.naturalNumer(array,size);

} break;

case 2:{

array=Calc.randomNumber(array,size);

} break;

case 3:{

array=Calc.naturalNumer(array,size);

array=Calc.reverseNumber(array);

}break;

default:

System.out.println("Not in range");

}

System.out.println("chosee \n\n 1.bubble sort \n\n 2.selection sort \n\n 3.beanchmark");

int alg = obj.nextInt();

switch (alg) {

case 1:

System.out.println("Bublesorting array");

bubblearray=Calc.bubblesort(array);

Calc.printarray(bubblearray);

break;

case 2:

selectionarray=Calc.selectionsort(array);

Calc.printarray(selectionarray);

break;

case 3:

//bubblesort

start=System.currentTimeMillis();

bubblearray=Calc.bubblesort(array);

stop=System.currentTimeMillis();

total=stop-start;

System.out.println("bubblesort time taken: "+total+" ms");

//selection sort

start=System.currentTimeMillis();

selectionarray=Calc.selectionsort(array);

stop=System.currentTimeMillis();

total=stop-start;

System.out.println("selectionsort time taken: "+total+" ms");

break;

default:

System.out.println("not in range");

break;

}

}

}

//calculator class

package corejavaproject;

public class Calc {

public static int [] naturalNumer( int[] array,int size){

for(int count=0;count<size;count++)

{

array[count]=count+1;

}

return array;

}

public static int[] randomNumber(int[] array, int size) {

for(int count=0;count<size;count++)

{

array[count]=(int) (Math.random()\*1000);

}

return array;

}

public static int [] reverseNumber( int[] array){

int index=0;

for(int count=array.length;count>0;count--)

{

array[index++]=count;

}

return array;

}

public static int[] bubblesort(int[] array) {

int n = array.length;

int temp = 0;

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

for(int j=1; j < (n-i); j++){

if(array[j-1] > array[j]){

//swap the elements!

temp = array[j-1];

array[j-1] = array[j];

array[j] = temp;

}

}

}

return array;

}

public static int[] selectionsort(int[] arr) {

for (int i = 0; i < arr.length - 1; i++)

{

int index = i;

for (int j = i + 1; j < arr.length; j++)

if (arr[j] < arr[index])

index = j;

int smallerNumber = arr[index];

arr[index] = arr[i];

arr[i] = smallerNumber;

}

return arr;

}

public static void printarray(int[] array){

for (int i = 0; i < array.length; i++) {

System.out.println(array[i]);

}

}

}

