package j1\_s\_p0001\_bubble\_sort;

import java.util.Random;

import java.util.Scanner;

/\*\*

\*

\* @author bravee06

\*/

public class J1\_S\_P0001\_Bubble\_Sort {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// 1. User enter a positive number for size of the array

int size = inputInt("Enter number of array:",0,Integer.MAX\_VALUE);

// 2. Create a random number array in range

int array[] = getRandomIntegerArray(size);

// 3. Display array before sort

display(array,"Unsorted array: ");

// 4. Bubble sort to sort ascending order

sortAscendingBubbbleSort(array);

// 5. Display array after sort

display(array,"Sorted array: ");

}

/\*

Function 1 - Display a screen to prompt users to input a positive decimal number

Param

msg: Sentence will notify on screen before user input integer number

min: number return must great than or equal min

max: number return must less than or equal max

Users run the program, display a screen to ask users to enter a positive decimal number and store value into a variable

If user enter a negative number require user enter a positive decimal number again

\*/

public static int inputInt(String msg, int min, int max) {

// Loop use to continue input number till return correct format, if catch a user exception or system exception

while (true) {

// If result value is not number then notify error number format for user and catch NumberFormatException to handle and continue input number

try {

// Display message notify user input number

System.out.println(msg);

// Use Scanner class to input and store value inputted into a new String variable

Scanner sc = new Scanner(System.in);

String result\_raw = sc.next();

// Parse String value to Int value

int result = Integer.parseInt(result\_raw);

// If result value less than min or great than max then notify error format for user and continue loop

if (result < min || result > max) {

System.out.println("You must enter number have range from " + min + " to " + max + " ! Again !");

continue;

}

// If not error then return result number

return result;

} catch (NumberFormatException e) {

System.out.println("You must enter number format !");

continue;

}

}

}

/\*

Function 2 - Display & sort array

\*/

// Generate random integer in number range for each array element

public static int[] getRandomIntegerArray(int size){

// Declear new array to store random array

int array[] = new int[size];

// Loop use to accessed from the first to last element of array

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

// number range is equal to size

int number\_range = size;

Random random = new Random();

int random\_number = random.nextInt(number\_range);

// assign random\_number to value of each value elements in array

array[i] = random\_number;

}

return array;

}

// Display array

public static void display(int array[],String msg){

System.out.print(msg);

int size = array.length;

// declare end\_bracket\_index to process problem close bracket of display array ( ... ] )

int end\_bracket\_index = size - 1;

// display open bracket

System.out.print("[");

// loop use to display element in the array

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

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

// if index of element not equal to end\_bracket\_index then add `, ` to display

if(i != end\_bracket\_index)

{

System.out.print(", ");

}

}

// display close bracket

System.out.print("]");

if(msg.equals("Unsorted array: ")){

System.out.println();

}

}

// Use Bubble Sort to sort by ascending number in array

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

int size = array.length;

// Loop use to accessed from the first to last element of array

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

/\*Loop use to accessed first unsorted element to the element

stand before the last unsorted element\*/

for(int j = 0; j < size - (i+1); j++){

/\* compare each pair adjacent elements

if reversed order then swap order

\*/

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

// use temp variable store value of first element

int temp = array[j];

// assign value of second element for first element

array[j] = array[j+1];

// assign value of temp element for second element

array[j+1] = temp;

}

}

}

return array;

}

}