**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**«ФАХОВИЙ КОЛЕДЖ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**ЛАБОРАТОРНА РОБОТА №8**

**Написання програм для сортування масивів даних**

***Роботу виконала студентка***

гр. КН-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Раделіцька С. І.

Дата виконання “22” листопада 2024\_\_

***Роботу прийняв викладач***

Селемонавічус А.А.

***Дані про залік***

Оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата захисту “\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024

Підпис викладача \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Львів – 2024**

**1. Мета роботи:**

|  |
| --- |
| Вивчити методи сортування масивів: обміном та вибором. А також застосування вбудованих методів для сортування масивів примітивних типів, об’єктів, списків. |

1. **Теоретичні відомості**

|  |
| --- |
| Питання вхідного контролю   1. Сортуванням називається такий процес перестановки елементів масиву, коли всі його елементи вистроюються по зростанню або спаданню. 2. У всьому масиві відшукуємо мінімальний елемент, міняємо його місцями з початковим. Потім в частині масиву, що залишилась (т.т. серед всіх елементів масиву крім першого) знову шукаємо мінімальний елемент, міняємо його місцями з другим елементом у масиві. І так далі. 3. Якщо попарно перевіряти два сусідніх елементи, то після проходу по всіх елементах масиву на останньому місці буде найбільший елемент масиву. Якщо ще раз пройтись по масиву то на передостанньому місці буде потрібний елемент. І так далі. |

1. **Інструмент, обладнання і прилади**

ПК, ОС Windows, JDK, Eclipse workspace.

**Завдання**

|  |
| --- |
| Використовуючи індивідуальні завдання записати програму. |

**Звіт по роботі**

|  |
| --- |
| //Example1  package lab8;  public class Example1 {  public static void main(String args[]) {    int[] a = {7, 6, 2, 4, 5};  for (int i = 0; i < a.length; i++) {    int min = a[i];  int imin = i;    for (int j = i+1; j < a.length; j++) {    if (a[j] < min) {  min = a[j];  imin = j;  }  }    if (i != min) {  int temp = a[i];  a[i] = a[imin];  a[imin] = temp;  }  System.out.println(a[i]);  }  }  }  Результат:    //Example2  package lab8;  public class Example2 {  public static void main (String args []) {    int [] a = {7, 6, 2, 4, 5};  for (int i = a.length - 1; i>0; i--) {    boolean sorted = true;    for (int j = 0; j<i; j++) {  if (a[j]>a[j+1]) {  int temp = a[j];  a[j] = a[j+1];  a[j+1] = temp;  sorted = false;  }  }    if (sorted) {  break;  }  }  for (int k = 0; k < a.length; k++) {  System.out.println(a[k]);  }  }  }  Результат:    //Example3  package lab8;  import java.util.Arrays;  public class Example3 {  public static void main(String args[]) {  int []intArr = {5, 4, 1, 0, 7};  String []strArr = {"S", "O", "F", "I", "A"};    Arrays.sort(intArr);  Arrays.sort(strArr);    System.out.println(Arrays.toString(intArr));  System.out.println(Arrays.toString(strArr));  }  }  Результат: |

**Відповіді на запитання вихідного контролю**

|  |
| --- |
| 1. У всьому масиві відшукуємо мінімальний елемент, міняємо його місцями з початковим. Потім в частині масиву, що залишилась (т.т. серед всіх елементів масиву крім першого) знову шукаємо мінімальний елемент, міняємо його місцями з другим елементом у масиві. І так далі. 2. Якщо попарно перевіряти два сусідніх елементи, то після проходу по всіх елементах масиву на останньому місці буде найбільший елемент масиву. Якщо ще раз пройтись по масиву то на передостанньому місці буде потрібний елемент. І так далі. 3. Самим зручним, простим та ефективним методом сортування масиву є метод sort() класу Arrays. Для цього варто імпортувати клас Arrays, та використовуючи команду Arrays.sort() відсортувати масив. |

**Висновки**

|  |
| --- |
| На цій лабораторній роботі я вивчила методи сортування масивів: обміном та вибором. А також застосування вбудованих методів для сортування масивів примітивних типів, об’єктів, списків. |