



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

**Лабораторна робота №6**  
із дисципліни «Основи програмування»  
**Тема: «Алгоритми сортування»**

Виконали:  
Студенти групи ІА-24  
Зелінський І.О.  
Криворучек В.С.  
Трасковський Т.І.

Перевірив:  
Колеснік Валерій Миколайович

1. Ознайомитись з алгоритмами сортування:

1.1. Сортування обміном (сортування бульбашкою, Bubble sort)

1.2. Сортування вибором (Selection sort)

1.3. Сортування включенням (сортування вставкою, Insertion sort)

2. Реалізувати два методи сортування відповідно до свого варіанту з таблиці 1.

6	byte[]	по спаданню	Insertion sort	Bubble sort
1	public class SortBubble {			<terminated> SortBubble [Java Application] C
2	public static void main(String[] args) {			Масив:
3	printResults(new byte[]{22, 1, 6, -3, 87, -77, 0, 0, -22, 1});			22 1 6 -3 87 -77 0 0 -22 1
4	printResults(new byte[] {5});			Відсортований масив:
5	printResults(new byte[] {-4, 8});			87 22 6 1 1 0 0 -3 -22 -77
6	printResults(new byte[] {1, 3, 5, 7});			Масив:
7	printResults(new byte[] {5, 4, 3, 0, -2, 7});			5
8	}			Відсортований масив:
9				5
10	public static byte[] sortBubble(byte[] arr){			Масив:
11	for (int i = arr.length; i>0; i--) {			-4 8
12	for (int j = 0; j<i-1; j++) {			Відсортований масив:
13	if (arr[j]<arr[j+1]) {			8 -4
14	byte k = arr[j];			Масив:
15	arr[j]=arr[j+1];			1 3 5 7
16	arr[j+1]= k;			Відсортований масив:
17	}			7 5 3 1
18	}			Масив:
19	}			5 4 3 0 -2 7
20	return arr;			Відсортований масив:
21	}			7 5 4 3 0 -2
22				
23	static void printResults(byte[] arr) {			
24	System.out.println("Масив:");			
25	for(byte i: arr) {			
26	System.out.print(i + " ");			
27	}			
28	System.out.println();			
29	System.out.println("Відсортований масив:");			
30	for(byte i: sortBubble(arr)) {			
31	System.out.print(i + " ");			
32	}			
33	System.out.println();			
34	}			
35	}			
1	public class SortInsertion{			<terminated> SortInsertion [Java Application]
2	public static void main(String[] args) {			Масив:
3	printResults(new byte[]{22, 1, 6, -3, 87, -77, 0, 0, -22, 1});			22 1 6 -3 87 -77 0 0 -22 1
4	printResults(new byte[] {5});			Відсортований масив:
5	printResults(new byte[] {-4, 8});			87 22 6 1 1 0 0 -3 -22 -77
6	printResults(new byte[] {1, 3, 5, 7});			Масив:
7	printResults(new byte[] {5, 4, 3, 0, -2, 7});			5
8	}			Відсортований масив:
9				5
10	public static byte[] sortInsertion(byte[] arr){			Масив:
11	for (int i = 1; i<arr.length; i++) {			-4 8
12	byte main = arr[i];			Відсортований масив:
13	int index = i-1;			8 -4
14	while((index>=0) && (main>arr[index])) {			Масив:
15	arr[index+1] = arr[index];			1 3 5 7
16	index=index-1;			Відсортований масив:
17	}			7 5 3 1
18	arr[index+1]=main;			Масив:
19	}			5 4 3 0 -2 7
20	return arr;			Відсортований масив:
21	}			7 5 4 3 0 -2
22				
23	static void printResults(byte[] arr) {			
24	System.out.println("Масив:");			
25	for(byte i: arr) {			
26	System.out.print(i + " ");			
27	}			
28	System.out.println();			
29	System.out.println("Відсортований масив:");			
30	for(byte i: sortInsertion(arr)) {			
31	System.out.print(i + " ");			
32	}			
33	System.out.println();			
34	}			
35	}			

Висновок: на цій лаб. роботі ми ознайомилися з основними простими алгоритмами сортування, їх перевагами та недоліками, деякі з них реалізували та перевірили на практиці.