МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет електроніки і комп’ютерних технологій

**Звіт**

Про виконання лабораторної роботи № 2

**«Прості методи сортування.»**

**Виконав:**

Студент групи ФЕП-14с

Грицюк Максим Олегович

**Перевірив:**

ас. Баран М. О.

Львів 2025

**Мета:** Опанувати прості методи сортування, такі як: сортування обміном;сортування вставкою; сортування вибором. А також закріпити знання, виконавши практичні завдання.

**Обладнання:**

Комп'ютер з встановленим програмним забезпеченням(Intellij IDEA Ultimate).

**Теоретичні відомості**

**Теоретичні відомості. Прості методи сортування**

Сортування (упорядкування) є однією з найбільш поширених

операцій обробки даних. Розрізняють методи внутрішнього та зовнішнього

сортування. Методи внутрішнього сортування застосовують до невеликих

об’ємів даних і вони не вимагають додаткової пам’яті. Методи

зовнішнього сортування застосовують до великих масивів даних, що

зберігаються на зовнішніх носіях.

Серед методів внутрішнього сортування розрізняють прості та

удосконалені методи.

Найбільш відомими простими методами сортування є:

-сортування обміном (бульбашкове сортування);

-сортування вставкою;

­­­­­-сортування вибором.

*Алгоритм сортування обміном (функція* SortBubble(…)*).*

1. Встановити лічильник ітерацій рівний одиниці. Поки лічильник не

досяг значення індексу останнього елементу масиву (цикл за

зростанням індексу):

1.1. Для елементів масиву: від останнього елементу до елементу з

індексом, рівному значенню лічильника ітерацій (цикл за спаданням

індексу):

1.1.1. Якщо поточний елемент є менший за наступний, то поміняти ці

елементи місцями.

*Алгоритм сортування вставкою (функція* SortInsertion(…)*).*

1. Встановити лічильник ітерацій рівний одиниці. Поки лічильник не

досяг значення індексу останнього елементу масиву (цикл за

зростанням індексу):

1.1. Занести елемент, що вставляється (перший в невідсортованій

ділянці – його індекс рівний лічильнику ітерацій) в допоміжну

змінну.

1.2. Для елементів масиву: від останнього елементу відсортованої

частини допоки не знайдено місце для вставки, або не досягнуто

початку масиву (цикл за спаданням індексу):

1.2.1. Якщо ключ поточного елементу є більший за ключ елементу,

що вставляється, то зсунути поточний елемент на одну позицію

вправо (пошук місця для вставки).

1.3. Вставити на місце останнього поточного елементу значення

елементу, що вставляється з додаткової змінної.

*Алгоритм сортування вибором (функція* SortSelection(…)*).*

1. Для елементів масиву: від першого до останнього елементів (цикл за

зростанням індексу):

1.1. Знайти елемент з мінімальним ключем (функція MinItem(…)) серед

елементів від поточного до останнього.

1.2. Переставити місцями знайдений елемент з мінімальним ключем з

поточним.

Обчислювальна складність простих алгоритмів сортування становить

*O*(*n*2 ) , де *n* – кількість елементів масиву, що обумовлено наявністю двох

вкладених циклів. Окрім того, у зв’язку з великою кількістю перестановок,

метод обміну вважається найгіршим і на практиці не використовується.

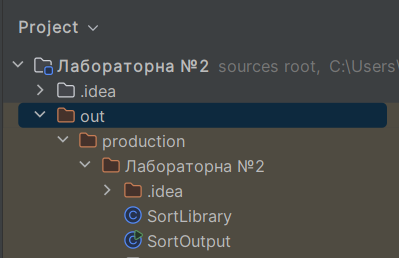
Серед вищеописаних алгоритмів найкращим вважається метод вставки,

однак для великих наборів даних доцільніше використовувати

удосконалені методи.

**Хід роботи**

**Завдання №1.** Створити нову бібліотеку з необхідними файлами(файли SortLibrary.java, SortOutput.java).

****

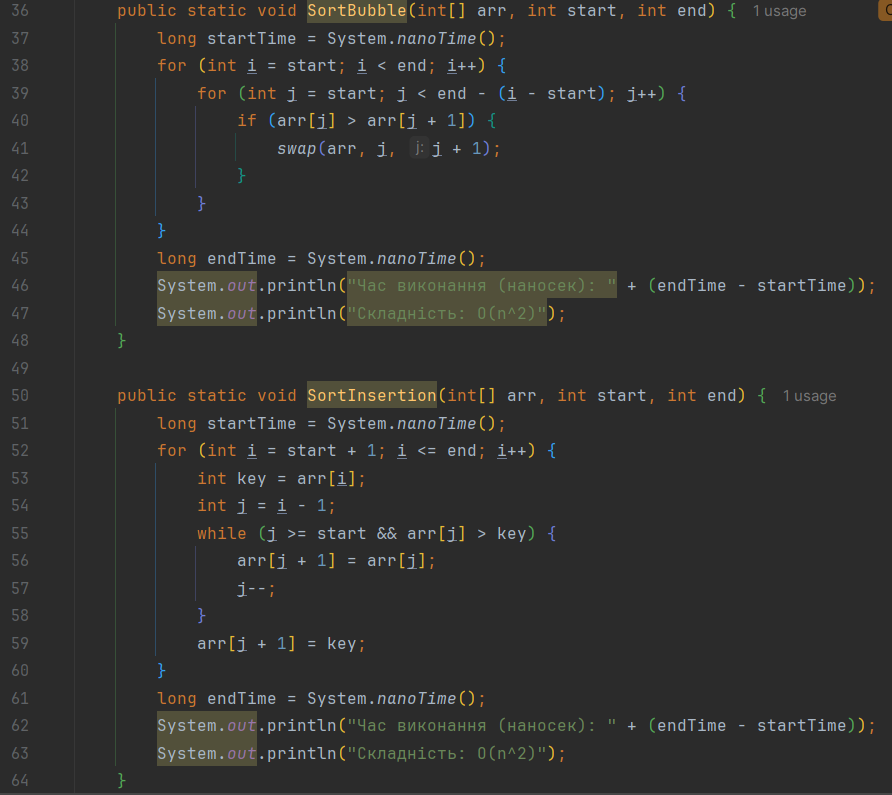
Створив нову бібліотеку, що складається з файлів “SortLibrary.java” та “SortOutput.java”.

**Завдання №2.** У бібліотеці Sort, згідно описаних вище алгоритмів, реалізувати

функції сортування методом обміну (бульбашки), вставки та вибору

(функції SortBubble(…), SortInsertion(…), SortSelection(…)).

Ці функції повинні приймати як параметри вказівник (ім’я) на масив.



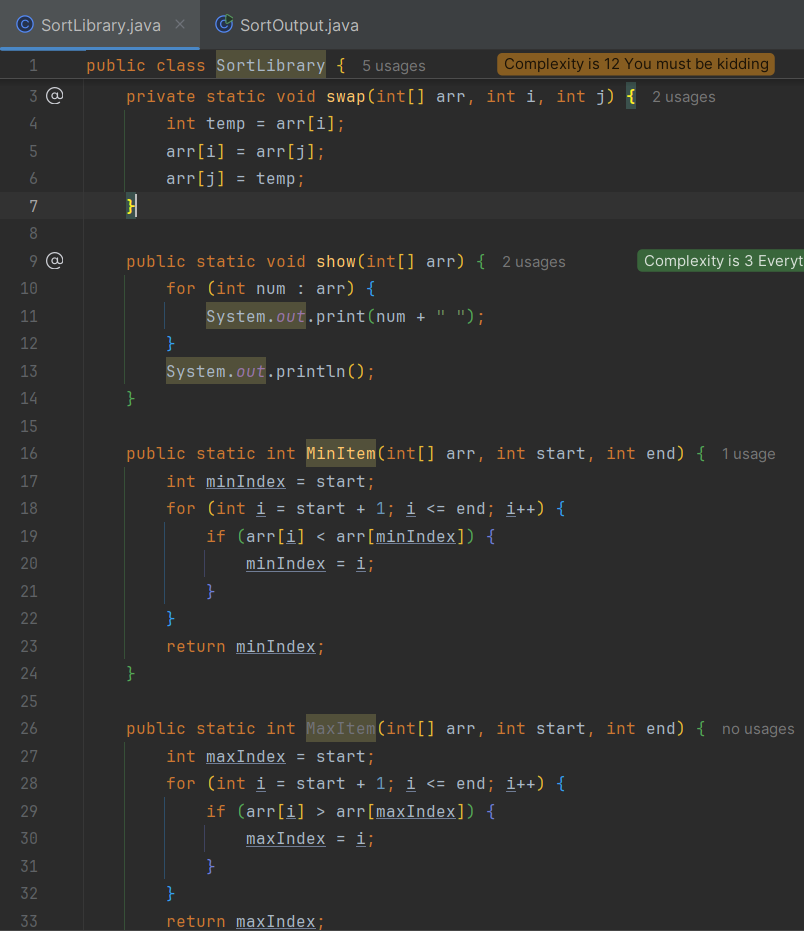
У файлі “SortLibrary.java” реалізував три методи сортування:

SortBubble() — сортування методом бульбашки. SortInsertion() — сортування методом вставки. SortSelection() — сортування методом вибору.

**Завдання №3.** В цій же бібліотеці запрограмувати допоміжні функції для

перестановки елементів масиву (swap(…)), відображення вмісту

масиву (show(…)) та пошуку елементу з мінімальним та

максимальним ключем у частині масиву (MinItem(…), MaxItem(…)).

У файлі SortLibrary.java оголосив допоміжні функції для роботи з масивами:

swap() — для обміну двох елементів масиву.

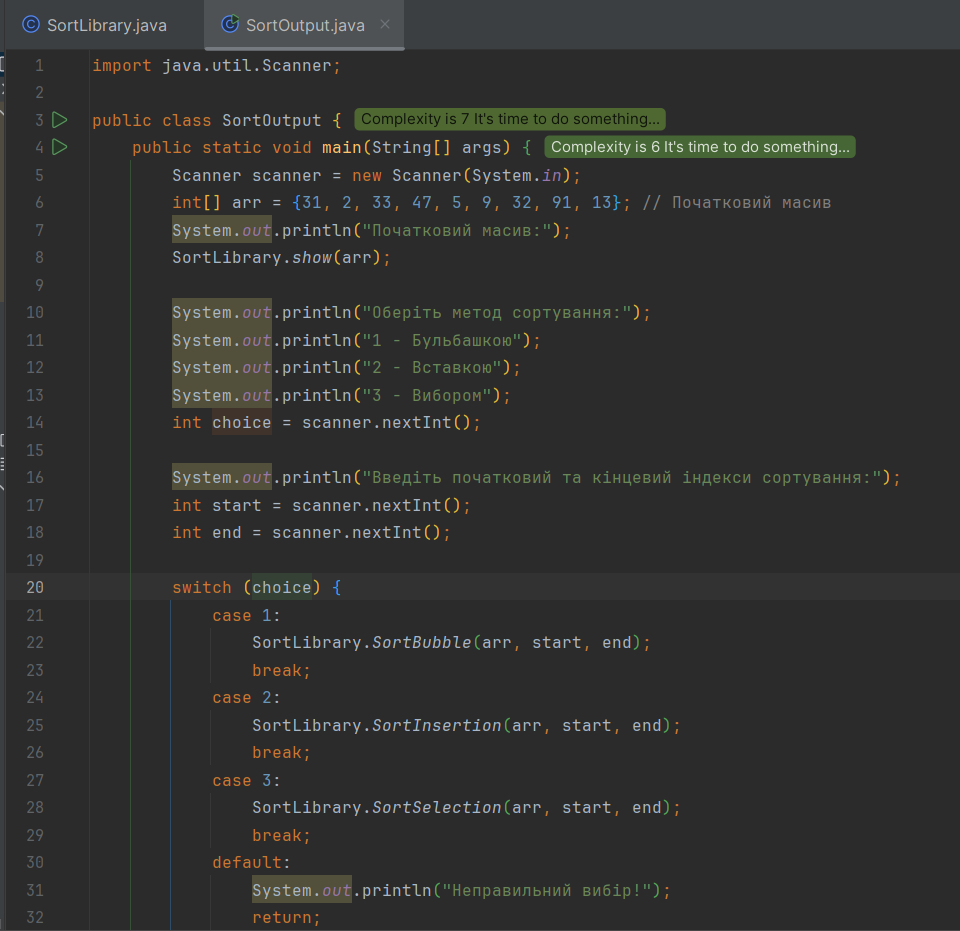
show() — для виведення масиву.

MinItem() — для знаходження мінімального елемента у заданій частині масиву; MaxItem() — для знаходження максимального елемента у заданій частині масиву.

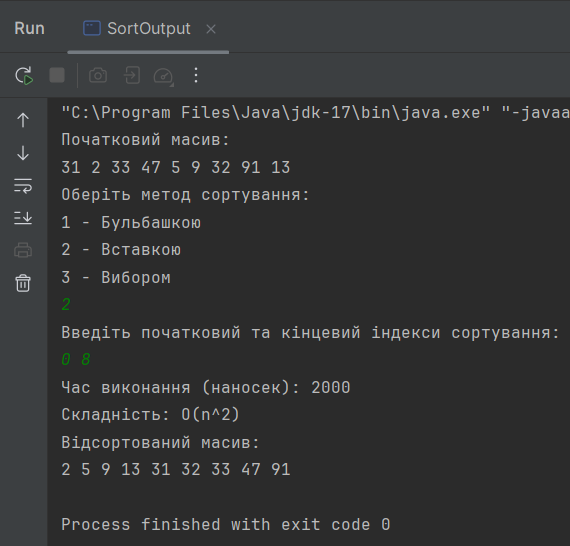
**Завдання №4.** Створити новий проект Lab\_2 до якого підключити бібліотеку Sort.

У функції main() проекту реалізувати меню для вибору методу сортування масиву з даними.

У файлі “SortOutput.java” реалізував консольне меню, яке дозволяє вибрати метод сортування та вказати його діапазон.



**Завдання №5.** Отримати від викладача завдання: масив даних відповідного типу. Продемонструвати викладачеві результат сортування цього масиву різними методами та для різних ділянок масиву.

****

**Висновок:** У ході виконання лабораторної роботи було реалізовано та досліджено три алгоритми сортування. Було додано можливість сортування частини масиву, що дозволяє застосовувати алгоритми для специфічних завдань. Реалізовані методи забезпечують правильне впорядкування елементів, що підтверджується тестовим прикладом. Робота допомогла закріпити знання з алгоритмів сортування та роботи з масивами в Java.