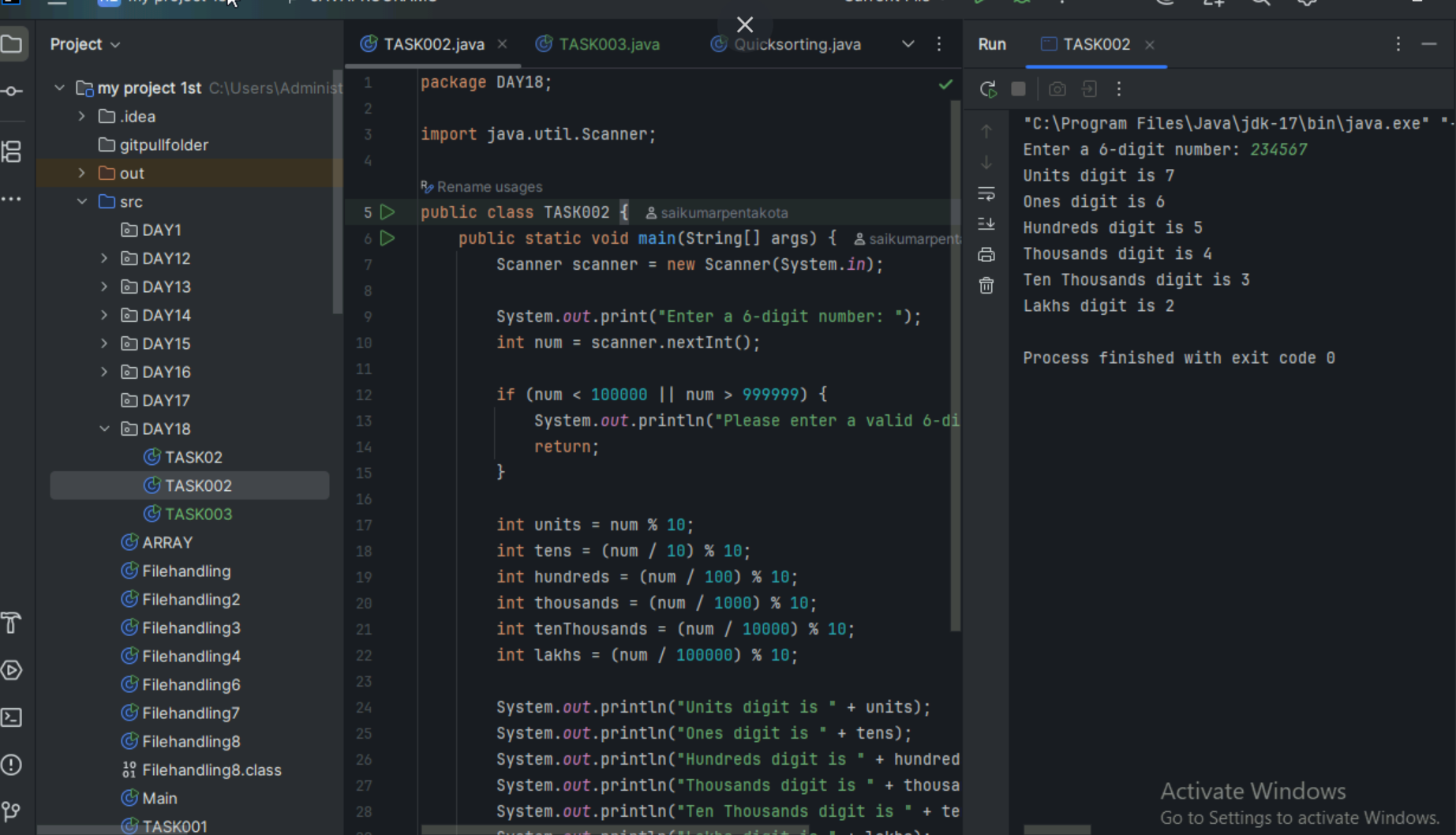
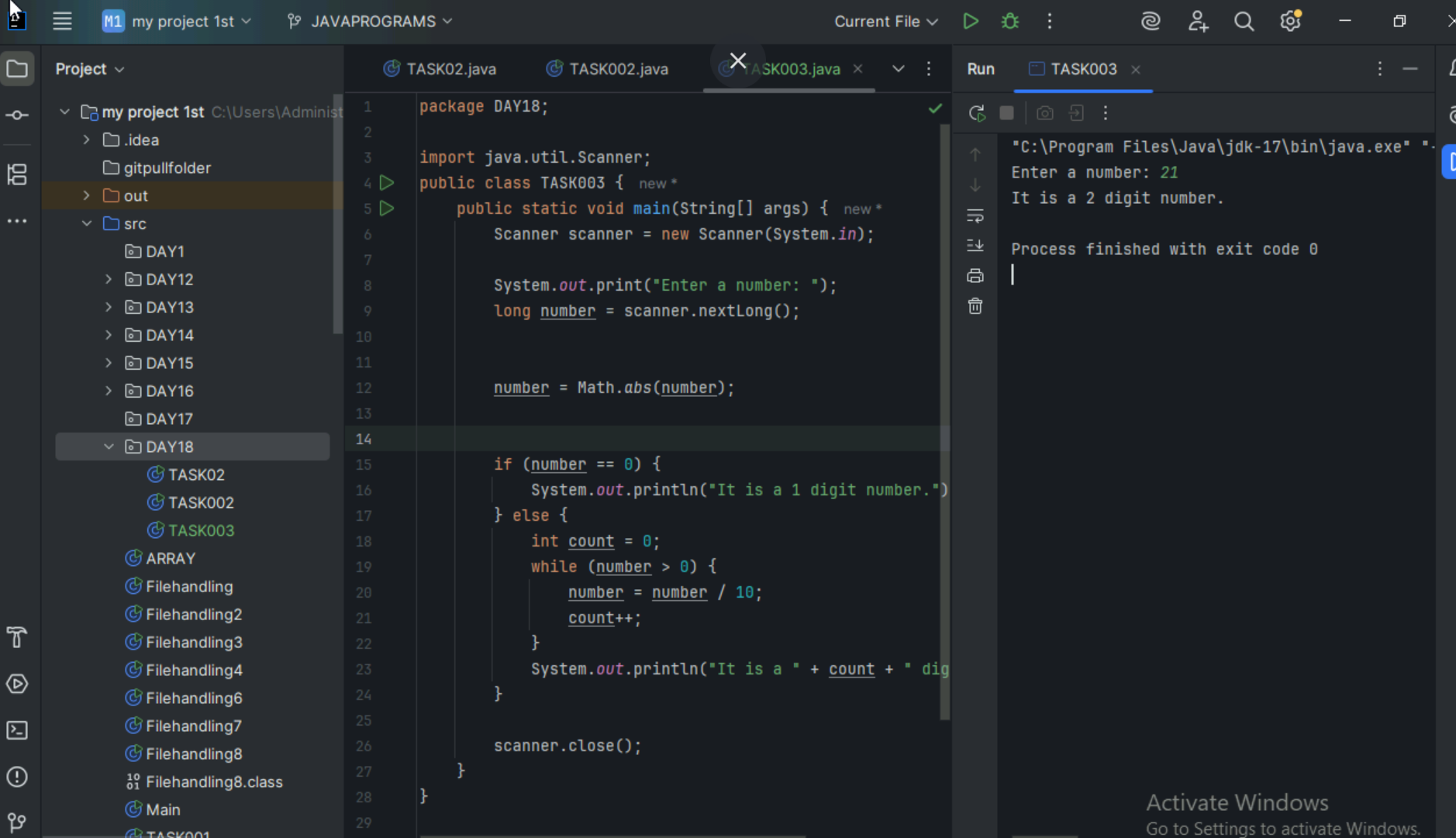
Day 18

Saikumar pentakota

TASK02



TASK03

TASK04

Heap sort has several practical applications:

1. Priority Queues

-Managing task scheduling in operating systems

- Handling event-driven simulations

- Processing requests based on priority levels

2. Memory Management

- Managing system resources

- Handling memory allocation/deallocation

- Garbage collection in programming languages

3. Data Analysis

- Finding k largest/smallest elements in a dataset

- Processing streaming data

- Real-time data processing

4. Graph Algorithms

- Dijkstra's shortest path algorithm

- Prim's minimum spanning tree algorithm

- Network flow algorithms

5. External Sorting

- Sorting large files that don't fit in memory

- Database operations

- File system organization

6. Gaming Applications

- AI decision making

- Game state management

- Pathfinding algorithms

7. System Applications

- Process scheduling in operating systems

- Printer queue management

- Network packet prioritization

TASK06

Certainly! Here's an explanation of how the Heap Sort algorithm works in 5 steps:

1. Build a Max Heap:

- Start with an unsorted array

- Transform the array into a max heap, where the largest element is at the root

- This is done by applying the heapify operation from the last non-leaf node up to the root

2. Swap Root with Last Element:

- Exchange the root (largest element) with the last element of the heap

- This places the largest element at the end of the array

3. Reduce Heap Size:

- Decrease the size of the heap by 1

- This effectively removes the sorted element from the heap

4. Heapify the Root:

- Apply the heapify operation on the root to ensure the heap property is maintained

- This brings the next largest element to the root

5. Repeat Steps 2-4:

- Continue this process until the heap size becomes 1

- With each iteration, the next largest element is placed at the end of the array

After these steps, the array will be sorted in ascending order. The time complexity of Heap Sort is O(n log n) for all cases, making it an efficient sorting algorithm for large datasets.

TASK10

