WEEK – 1

Implementing the task management system

Nipuna Amanapu

[namanapu@gitam.in](mailto:namanapu@gitam.in)

superset id : 6432842

**Exercise 5: Task Management System**

**Scenario:**

You are developing a task management system where tasks need to be added, deleted, and traversed efficiently.

**Steps:**

1. **Understand Linked Lists:**
   * Explain the different types of linked lists (Singly Linked List, Doubly Linked List).
2. **Setup:**
   * Create a class **Task** with attributes like **taskId**, **taskName**, and **status**.
3. **Implementation:**
   * Implement a singly linked list to manage tasks.
   * Implement methods to **add**, **search**, **traverse**, and **delete** tasks in the linked list.
4. **Analysis:**
   * Analyze the time complexity of each operation.
   * Discuss the advantages of linked lists over arrays for dynamic data.

Solution:

1. Task.java

public class Task {

int taskId;

String taskName;

String status;

public Task(int taskId, String taskName, String status) {

this.taskId = taskId;

this.taskName = taskName;

this.status = status;

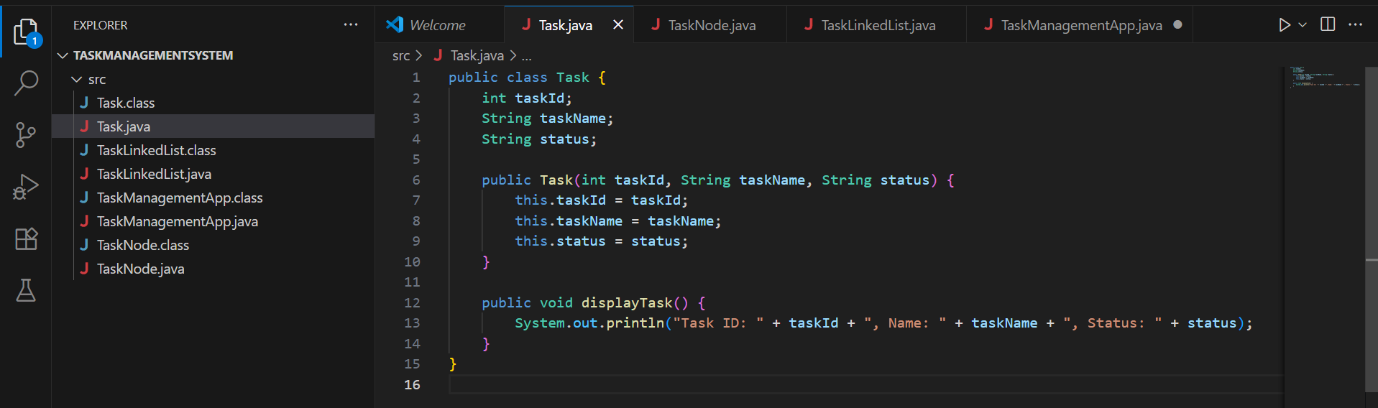
}

public void displayTask() {

System.out.println("Task ID: " + taskId + ", Name: " + taskName + ", Status: " + status);

}

}



1. TaskNode.java

public class TaskNode {

Task task;

TaskNode next;

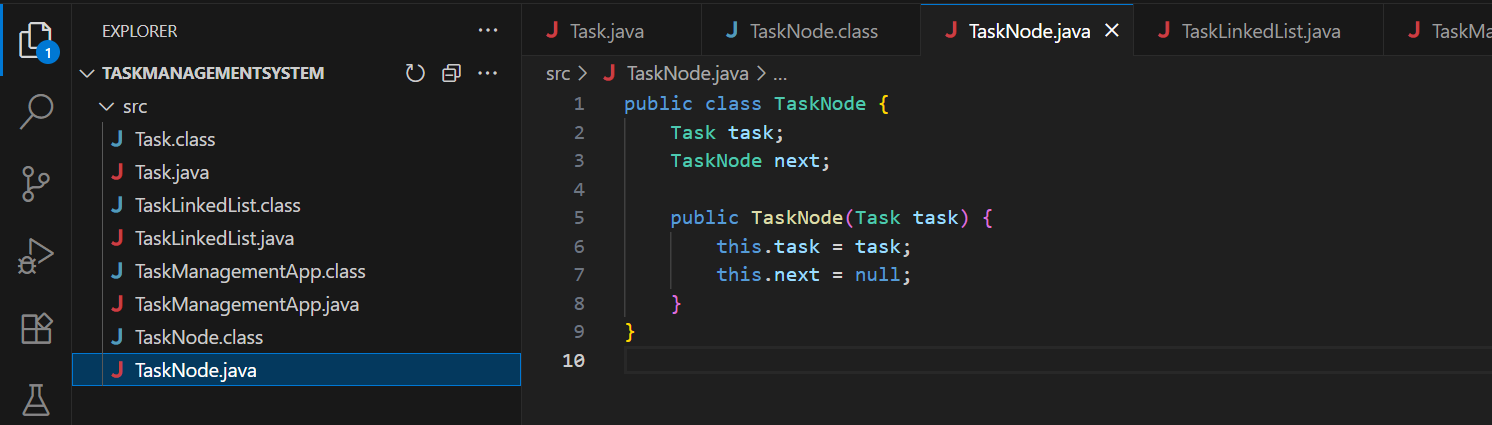
public TaskNode(Task task) {

this.task = task;

this.next = null;

}

}



1. TaskLinkedList.java

public class TaskLinkedList {

private TaskNode head;

public void addTask(Task task) {

TaskNode newNode = new TaskNode(task);

if (head == null) {

head = newNode;

} else {

TaskNode current = head;

while (current.next != null) {

current = current.next;

}

current.next = newNode;

}

System.out.println("Task added successfully.");

}

public void displayTasks() {

if (head == null) {

System.out.println("No tasks available.");

return;

}

TaskNode current = head;

while (current != null) {

current.task.displayTask();

current = current.next;

}

}

public void searchTask(int taskId) {

TaskNode current = head;

while (current != null) {

if (current.task.taskId == taskId) {

System.out.println("Task found:");

current.task.displayTask();

return;

}

current = current.next;

}

System.out.println("Task with ID " + taskId + " not found.");

}

public void deleteTask(int taskId) {

if (head == null) {

System.out.println("Task list is empty.");

return;

}

if (head.task.taskId == taskId) {

head = head.next;

System.out.println("Task deleted.");

return;

}

TaskNode current = head;

while (current.next != null) {

if (current.next.task.taskId == taskId) {

current.next = current.next.next;

System.out.println("Task deleted.");

return;

}

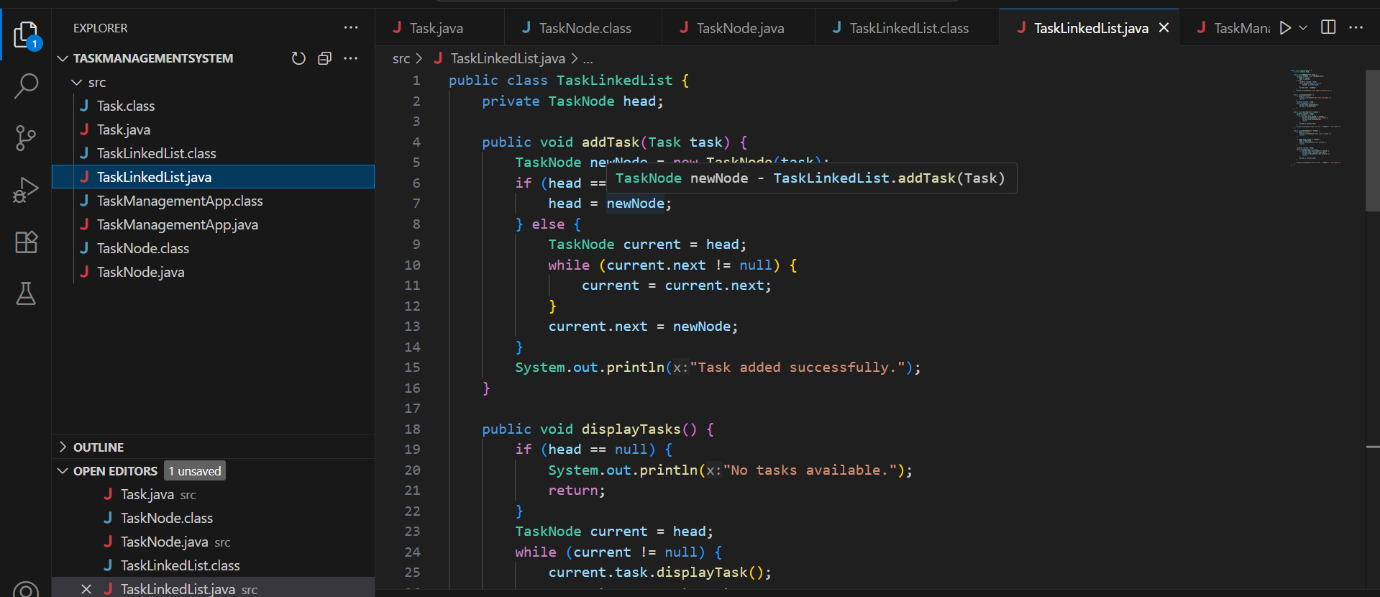
current = current.next;

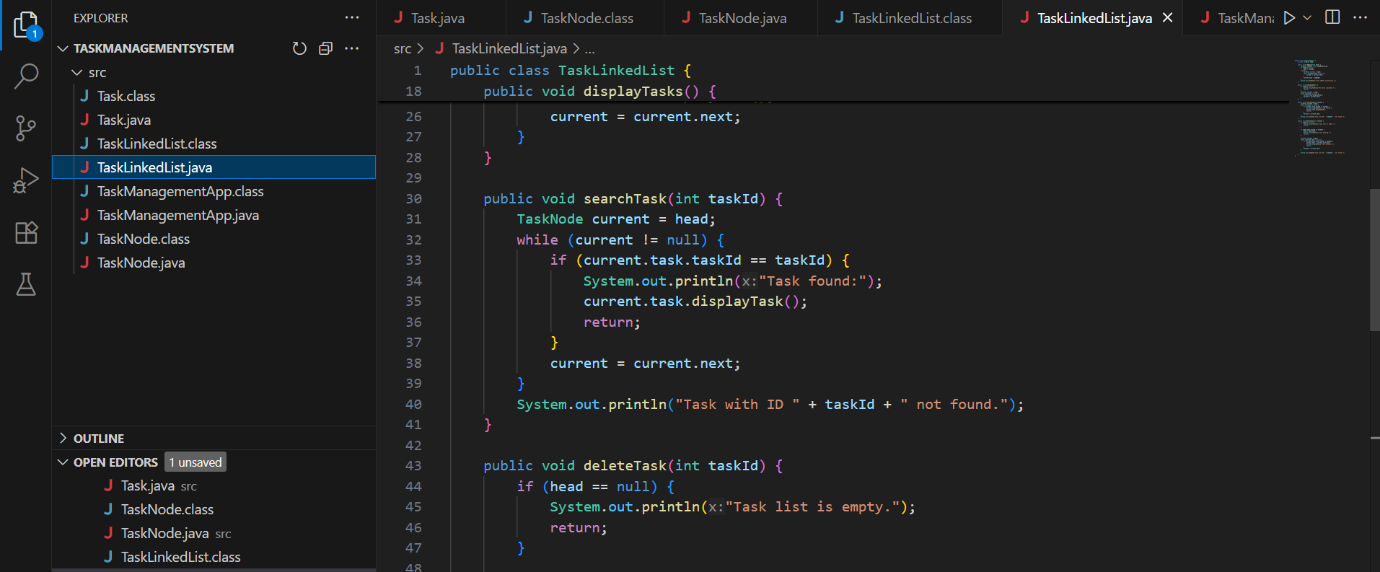
}

System.out.println("Task with ID " + taskId + " not found.");

}

}





1. TaskManagementApp.java

import java.util.Scanner;

public class TaskManagementApp {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

TaskLinkedList taskList = new TaskLinkedList();

while (true) {

System.out.println("\n--- Task Management System ---");

System.out.println("1. Add Task");

System.out.println("2. Display All Tasks");

System.out.println("3. Search Task by ID");

System.out.println("4. Delete Task by ID");

System.out.println("5. Exit");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

scanner.nextLine();

switch (choice) {

case 1:

System.out.print("Enter Task ID: ");

int id = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter Task Name: ");

String name = scanner.nextLine();

System.out.print("Enter Task Status: ");

String status = scanner.nextLine();

Task newTask = new Task(id, name, status);

taskList.addTask(newTask);

break;

case 2:

taskList.displayTasks();

break;

case 3:

System.out.print("Enter Task ID to search: ");

int searchId = scanner.nextInt();

taskList.searchTask(searchId);

break;

case 4:

System.out.print("Enter Task ID to delete: ");

int deleteId = scanner.nextInt();

taskList.deleteTask(deleteId);

break;

case 5:

System.out.println("Exiting Task Management System.");

scanner.close();

return;

default:

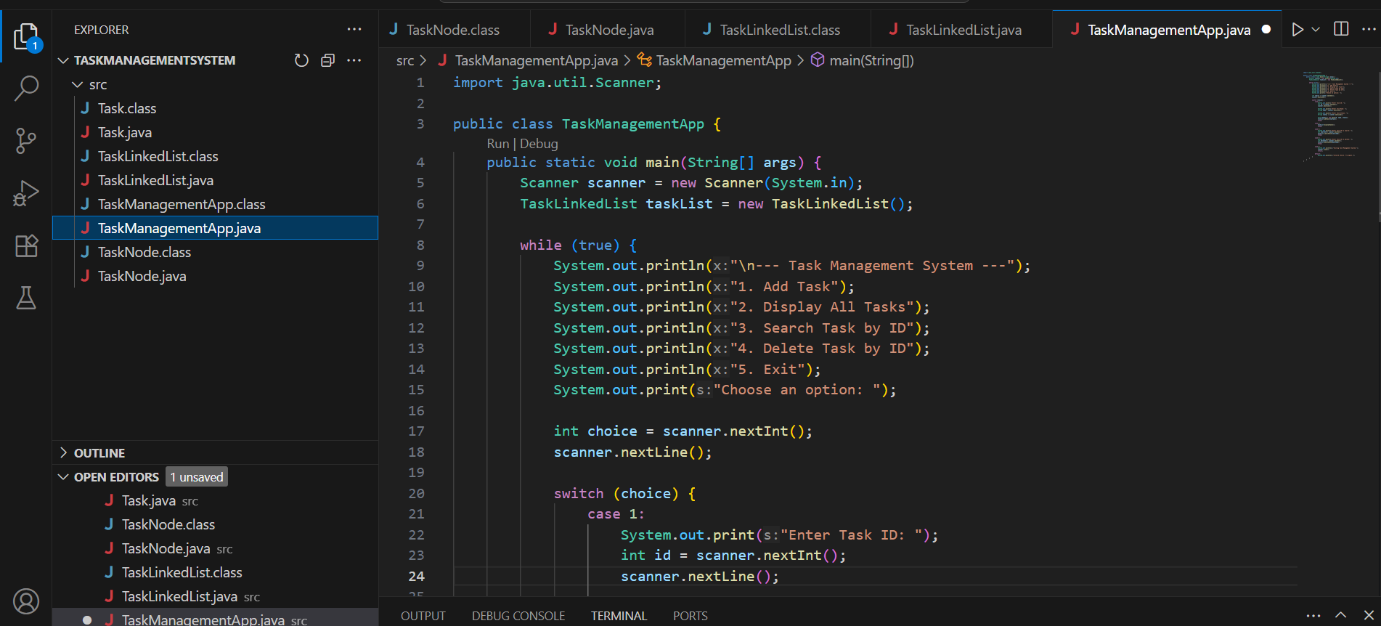
System.out.println("Invalid choice. Try again.");

}

}

}

}



Output:

