```
1 /* PRACTICAL-9: Mini Project - Implementation using above Data Structure.
 3 Name: Angat Shah
 4 Enrollment No: 202203103510097
 5 Branch: B.Tech Computer Science and Engineering */
 6
 7 // CODE:
 8
 9 import java.util.ArrayList;
10 import java.util.List;
11 import java.util.Scanner;
12
13 public class Main {
14
     public static void main(String[] args) {
15
       Scanner scanner = new Scanner(System.in);
16
       while (true) {
17
          System.out.println("Select a scheduling algorithm:");
          System.out.println("1. FCFS");
18
19
          System.out.println("2. SJF");
20
          System.out.println("3. SRT");
21
          System.out.println("4. PSN");
22
          System.out.println("5. PSP");
23
          System.out.println("6. RR");
24
          System.out.println("0. Exit");
25
          System.out.print("Enter your choice: ");
26
27
          int choice = scanner.nextInt();
28
          scanner.nextLine(); // Consume newline
29
30
          switch (choice) {
31
            case 1:
32
              System.out.println("-----");
              runAlgorithm(new FirstComeFirstServe());
33
34
              break;
35
            case 2:
              System.out.println("-----");
36
37
              runAlgorithm(new ShortestJobFirst());
              break;
38
39
            case 3:
              System.out.println("-----");
40
41
              runAlgorithm(new ShortestRemainingTime());
42
              break;
43
            case 4:
              System.out.println("-----");
44
45
              runAlgorithm(new PriorityNonPreemptive());
46
              break;
47
            case 5:
              System.out.println("-----");
48
49
              runAlgorithm(new PriorityPreemptive());
50
              break;
51
            case 6:
              System.out.println("-----");
52
53
              runRoundRobin();
54
              break;
55
            case 0:
56
              System.out.println("Exiting...");
57
              return;
58
            default:
```

```
59
                 System.out.println("Invalid choice. Please enter a valid option.");
 60
            }
 61
         }
       }
 62
 63
       public static void runAlgorithm(CPUScheduler scheduler) {
 64
         addProcesses(scheduler);
 65
         scheduler.process();
 66
         display(scheduler);
 67
       }
 68
       public static void runRoundRobin() {
 69
         CPUScheduler rr = new RoundRobin();
 70
         addProcesses(rr);
 71
         System.out.println("Enter time quantum for Round Robin: ");
 72
         Scanner scanner = new Scanner(System.in);
 73
         int timeQuantum = scanner.nextInt();
 74
         rr.setTimeQuantum(timeQuantum);
 75
         rr.process();
 76
         display(rr);
 77
       }
 78
       public static void addProcesses(CPUScheduler scheduler) {
 79
         Scanner scanner = new Scanner(System.in);
 80
         System.out.println("Enter number of processes: ");
 81
         int numProcesses = scanner.nextInt();
 82
         scanner.nextLine(); // Consume newline
         for (int i = 0; i < numProcesses; i++) {
 83
            System.out.println("Enter details for process" + (i + 1) + ":");
 84
 85
            System.out.print("Process Name: ");
 86
            String processName = scanner.nextLine();
 87
            System.out.print("Arrival Time: ");
            int arrivalTime = scanner.nextInt();
 88
 89
            System.out.print("Burst Time: ");
 90
            int burstTime = scanner.nextInt();
 91
            scanner.nextLine(); // Consume newline
 92
            scheduler.add(new Row(processName, arrivalTime, burstTime));
 93
         }
 94
       }
 95
       public static void display(CPUScheduler object) {
 96
         System.out.println("Process\tAT\tBT\tWT\tTAT");
 97
         for (Row row : object.getRows()) {
            System.out.println(row.getProcessName() + "\t" + row.getArrivalTime() + "\t"
 98
99
                 + row.getBurstTime() + "\t" + row.getWaitingTime() + "\t" + row.getTurnaroundTime());
100
         System.out.println();
101
         for (int i = 0; i < object.getTimeline().size(); i++) {
102
103
            List<Event> timeline = object.getTimeline();
104
            System.out.print(timeline.get(i).getStartTime() + "(" + timeline.get(i).getProcessName() + ")");
105
            if (i == object.getTimeline().size() - 1) {
106
              System.out.print(timeline.get(i).getFinishTime());
107
108
         }
109
         System.out.println("\n\nAverage WT: " + object.getAverageWaitingTime() + "\nAverage TAT: "
110
              + object.getAverageTurnAroundTime());
111
112 }
```