```
package ass5preemptiveprioruty;
import java.util.Scanner;

public class ass5preemptiveprioruty {

   public static void main(String[] args) {

     System.out.println("*** Priority Scheduling (Preemptive) ***");
     System.out.print("Enter Number of Process: ");
     Scanner sc = new Scanner(System.in);
     int n = sc.nextInt();
     int process[] = new int[n];
     int arrivaltime[] = new int[n];
     int burstTime[] = new int[n];
     int completionTime[] = new int[n];
     int priority[] = new int[n + 1];
     int TAT[] = new int[n];
     int waitingTime[] = new int[n];
     int burstTimecopy[] = new int[n];
     int min = 0, count = 0;
     int temp, time = 0, end;
     double avgWT = 0, avgTAT = 0;
     for (int i = 0; i < n; i++) {
       process[i] = (i + 1);
       System.out.println("");
       System.out.print("Enter Arrival Time for processor " + (i + 1) + ":");
       arrivaltime[i] = sc.nextInt();
       System.out.print("Enter Burst Time for processor " + (i + 1) + " : ");
       burstTime[i] = sc.nextInt();
       System.out.print("Enter Priority for " + (i + 1) + " process: ");
       priority[i] = sc.nextInt();
     }
     for (int i = 0; i < n - 1; i++) {
       for (int j = i + 1; j < n; j++) {
         if (arrivaltime[i] > arrivaltime[j]) {
           temp = process[i];
           process[i] = process[j];
           process[j] = temp;
           temp = arrivaltime[j];
           arrivaltime[j] = arrivaltime[i];
           arrivaltime[i] = temp;
           temp = priority[j];
           priority[j] = priority[i];
           priority[i] = temp;
           temp = burstTime[j];
           burstTime[j] = burstTime[i];
           burstTime[i] = temp;
         }
         if (arrivaltime[i] == arrivaltime[j] && priority[j] > priority[i]) {
           temp = process[i];
           process[i] = process[j];
           process[j] = temp;
           temp = arrivaltime[j];
           arrivaltime[j] = arrivaltime[i];
           arrivaltime[i] = temp;
```

```
          temp = priority[j];
          priority[j] = priority[i];
          priority[i] = temp;
          temp = burstTime[j];
          burstTime[j] = burstTime[i];
          burstTime[i] = temp;
        }
      }
    }
    System.arraycopy(burstTime, 0, burstTimecopy, 0, n);
    priority[n] = 999;
    for (time = 0; count != n; time++) {
      min = n;
      for (int i = 0; i < n; i++) {
        if (arrivaltime[i] <= time && priority[i] < priority[min] && burstTime[i] > 0)
          min = i;
      }
      burstTime[min]--;
      if (burstTime[min] == 0) {
        count++;
        end = time + 1;
        completionTime[min] = end;
        waitingTime[min] = end - arrivaltime[min] - burstTimecopy[min];
        TAT[min] = end - arrivaltime[min];
      }
    }

    for (int i = 0; i < n; i++) {
      avgTAT += TAT[i];
      avgWT += waitingTime[i];
    }
    System.out.println("\n*** Priority Scheduling (Preemptive) ***");
    System.out.println("Processor\tArrival time\tBrust time\tCompletion Time\t\tTurn around time\tWaiting ti
me");
    System.out.println(
        "-------------------------------------------------------------------------------------------------------");
    for (int i = 0; i < n; i++) {
      System.out.println("P" + process[i] + "\t\t" + arrivaltime[i] + "ms\t\t" + burstTimecopy[i] + "ms\t\t"
          + completionTime[i] + "ms\t\t\t" + TAT[i] + "ms\t\t\t" + waitingTime[i] + "ms");

    }
    avgWT /= n;
    avgTAT /= n;
    System.out.println("\nAverage Wating Time: " + avgWT);
    System.out.println("Average Turn Around Time: " + avgTAT);

    sc.close();
  }
 }
/*
*** Priority Scheduling (Preemptive) ***
Enter Number of Process: 4

Enter Arrival Time for processor 1:0
Enter Burst Time for processor 1 : 5
```

Enter Priority for 1 process: 10

Enter Arrival Time for processor 2:1
Enter Burst Time for processor 2 : 4
Enter Priority for 2 process: 20

Enter Arrival Time for processor 3:2
Enter Burst Time for processor 3 : 2
Enter Priority for 3 process: 30

Enter Arrival Time for processor 4:4
Enter Burst Time for processor 4 : 1
Enter Priority for 4 process: 40

*** Priority Scheduling (Non Preemptive) ***

Processor	Arrival time	Brust time	Completion Time	Turn around time	Waiting time
P1	0ms	5ms	5ms	5ms	0ms
P2	1ms	4ms	9ms	8ms	4ms
P3	2ms	2ms	11ms	9ms	7ms
P4	4ms	1ms	12ms	8ms	7ms

Gantt chart = p1 p2 p3 p3 p4 p2 p1
0 1 2 3 4 5 8 12

criteria = "Priority"
Mode = "preemptuve"
TAT = CT - AT
WT = TAT - BT
Average Wating Time: 4.5
Average Turn Around Time: 7.5
*/