SJFNonPreemptive.java

```
import java.util.Scanner;

class Process {
   int pid, at, bt, ct, tat, wt;
}

public class SJFNonPreemptive {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter the number of processes: ");
      int n = sc.nextInt();

      Process[] p = new Process[20];
      for (int i = 0; i < n; i++) p[i] = new Process();

      System.out.println("Enter Arrival Time for " + n + " processes:");
      for (int i = 0; i < n; i++) {
         System.out.print("P" + (i + 1) + ": ");
         p[i].at = sc.nextInt();
         p[i].pid = i + 1;
      }

      System.out.println("Enter Burst Time for " + n + " processes:");
      for (int i = 0; i < n; i++) {
         System.out.print("P" + (i + 1) + ": ");
         p[i].bt = sc.nextInt();
      }

      // Selection sort by Burst Time (stable for ties)
```

```
for (int i = 0; i < n - 1; i++) {
    int m = i;
    for (int j = i + 1; j < n; j++) {
        if (p[j].bt < p[m].bt)
            m = j;
    }
    if (m != i) {
        Process temp = p[i];
        p[i] = p[m];
        p[m] = temp;
    }
}

int t = 0;
double sumTAT = 0, sumWT = 0;

// Compute Completion Time, Turnaround Time, and Waiting Time
for (int i = 0; i < n; i++) {
    if (p[i].at > t)
        t = p[i].at; // CPU idle until process arrives
    t += p[i].bt;
    p[i].ct = t;
    p[i].tat = p[i].ct - p[i].at;
    p[i].wt = p[i].tat - p[i].bt;
    sumTAT += p[i].tat;
    sumWT += p[i].wt;
}

// Display results
System.out.println("\n=============================================");
System.out.println("Process\tAT\tBT\tCT\tTAT\tWT");
```

```
        System.out.println("================================================");
        for (int i = 0; i < n; i++) {
            System.out.println("P" + p[i].pid + "\t" + p[i].at + "\t" + p[i].bt + "\t" +
                    p[i].ct + "\t" + p[i].tat + "\t" + p[i].wt);
        }
        System.out.println("================================================");
        System.out.printf("Average Turnaround Time: %.2f\n", (sumTAT / n));
        System.out.printf("Average Waiting Time:    %.2f\n", (sumWT / n));

        sc.close();
    }
}
```