**class** ProcessData {

**public** **int** arriveTime;

**public** **int** serviceTime;

**public** **int** finishTime;

**public** **int** turnTime;

**public** ProcessData(**int** arriveTime, **int** serviceTime) {

**this**.arriveTime = arriveTime;

**this**.serviceTime = serviceTime;

}

**public** String toString() {

**return** arriveTime + "\t" +

serviceTime + "\t" +

finishTime + "\t" +

turnTime ;

}

}

**public** **class** Process {

**public** **static** **double** *avgTotalTime*;// 平均周转总时间

**public** **static** **void** main(String[] args) {

ProcessData[] processData = **new** ProcessData[4];

// 定义四个进程

processData[0] = **new** ProcessData(0, 20); // 进程0

processData[1] = **new** ProcessData(5, 15); // 进程1

processData[2] = **new** ProcessData(10, 5); // 进程2

processData[3] = **new** ProcessData(15, 10); // 进程3

*SJF*(processData);

}

**private** **static** **void** SJF(ProcessData[] processData) {

**int** preFinished = 0; // 前一个进程的完成时间即下一个进程的开始时间

*avgTotalTime* = 0; // 平均周转时间

**for** (**int** i = 0; i < processData.length; i++) {

processData[i].finishTime = 0; // 设置完成时间为0

processData[i].turnTime = 0; // 设置周转时间为0

processData[i].powerTime = 0; // 设置平均周转时间为0

}

**int** number = 0; // 定义进程序号

// 定义双层for循环用于比较进程的完成时间和运行时间

**for** (**int** i = 0; i < processData.length; i++) {

**int** min = 10000;

**for** (**int** j = 0; j < processData.length; j++) {

**if** (processData[j].serviceTime < min && processData[j].arriveTime <= preFinished && processData[j].finishTime == 0) {

min = processData[j].serviceTime; // 将目前服务时间最短的进程的运行时间赋值给作业的最小完成时间

number = j; // 将下一个进行调度的进程序号赋值给number

}

}

processData[number].finishTime = preFinished + processData[number].serviceTime; // 当前进程的完成时间等于上一个进程的完成时间加当前进程的运行时间

preFinished = processData[number].finishTime; // 将上一个进程的完成时间赋值为当前进程的完成时间

processData[number].turnTime = processData[number].finishTime - processData[number].arriveTime; // 周转时间 = 完成时间 - 到达时间

}

System.***out***.println("短作业优先算法：");

*Display*(processData);

}

**private** **static** **void** Display(ProcessData[] processData) {

System.***out***.println("到达时间\t" + "运行时间\t" + "完成时间\t" + "周转时间\t");

**for** (**int** i = 0; i < processData.length; i++) {

System.***out***.println(processData[i]);

*avgTotalTime* += processData[i].turnTime; // 求总周转时间，此时avgTotalTime中存储的值为总周转时间

}

*avgTotalTime* = *avgTotalTime* / processData.length; // 平均周转时间

System.***out***.println("平均周转时间：" + *avgTotalTime);*

}