

```

package cpu;

import javax.swing.JRadioButton;

import Algorithm.FCFS;
import Algorithm.MyAlgorithm;
import Algorithm.Priority1;
import Algorithm.Priority2;
import Algorithm.RR;
import Algorithm.SJF;
import Algorithm.STRF;
import Items.Job;
import Items.Queue;

/**
 * This class is responsible for initiating, selecting the algorithm
 * and doing the step work for the simulation
 */
public class Simulation extends Face {

    private static MyAlgorithm myAlgorithm; // object used for polymorphism
    public static int Time; // current time of the simulation
    public static JRadioButton AlgorithmType ; // default algoithm type
    public static int Quantum = 2; // quantum time for round robin algorithm
    public static boolean Finished = false; // show that the simulation is finished
    public static boolean Stoped = true; // show that the simulation is stoped

    /**
     * reset the simulation
     */
    public static void reset()
    {
        Time = 0; // reset the simulation time
        Finished = false; // simulation is not finished
    }

    /**
     * @return the current ready queue of the working algorithm
     */
    public static Queue getReadyQueue()
    {
        return myAlgorithm.getReadyQueue();
    }

    /**
     * let the selected algorithm finish a step
     * @return the current job worked by the algorithm
     */

```

```

public static Job workStep()
{
    Job job;
    if(Time == 0) {selectAlgorithm();} // select and init the algorithm
    job = myAlgorithm.nextStep(Time);
    if(myAlgorithm.isFinished()){Finished = true;}
    return job;
}

/**
 * select and initiate the selected algorithm
 */
private static void selectAlgorithm()
{
    if(AlgorithmType == FCFS){myAlgorithm = new FCFS(MainQueue.get());} // first come
first served
    else if(AlgorithmType == SJF) {myAlgorithm = new SJF(MainQueue.get());} // shortest job first
    else if(AlgorithmType == Priority1) {myAlgorithm = new Priority1(MainQueue.get());} // priority non-
preemptive
    else if(AlgorithmType == STRF) {myAlgorithm = new STRF(MainQueue.get());} // shortest time
remaining first
    else if(AlgorithmType == Priority2) {myAlgorithm = new Priority2(MainQueue.get());} // priority
preemptive
    else if(AlgorithmType == RR) {myAlgorithm = new RR(MainQueue.get() , Quantum);} // Round
Robin
}
}

```