```
import java.util.LinkedList;

public class JobTable
{
    /**
     * VARIABLES*****************************************************************
     */
    // Table will store all jobs that have entered
    static LinkedList<Job> table;

    /**
     * CONSTRUCTOR***************************************************************
     */
    JobTable ()
    {
        table = new LinkedList<Job>();
    }

    /**
     * PUBLIC METHODS************************************************************
     */

    public static void add (int[] p)
    {
        Job newJob = new Job(p);
        table.add(newJob);
    }

    public static void clearAddress (int jobID)
    {
        if (!table.get(jobID-1).inMemory)
        {
            table.get(jobID - 1).address = -1;
        }
    }

    public static void clearIO (int jobID)
    {
        table.get(jobID - 1).pendingIO = 0;
    }

    public static int decrementIO (int jobID)
    {
        Job decJob = table.get(jobID - 1);
        decJob.pendingIO--;
        // System.out.println("-JobTable decrements I/O");
        // System.out.println("--Job# " + decJob.idNum + " has " + decJob.pendingIO + " i/o
requests");
        return decJob.pendingIO;
    }

    public static boolean doingIO(int jobID)
    {
        if (jobID != -1) {
            return table.get(jobID - 1).latched;
        }
        else {
```

```
        return false;
    }
}

public static int getAddress (int jobID)
{
    return table.get(jobID - 1).address;
}

public static int getCurrentCPUTime (int jobID)
{
    return table.get(jobID - 1).currentCPUTime;
}

public static int getDirection (int jobID)
{
    return table.get(jobID - 1).direction;
}

public static int getIO (int jobID)
{
    return table.get(jobID - 1).pendingIO;
}

public static int getMaxCPUTime (int jobID)
{
    return table.get(jobID - 1).maxCPUTime;
}

public static int getPriorityTime (int jobID)
{
    return table.get(jobID -1).priorityTime;
}

public static int getSize (int jobID)
{
    return table.get(jobID -1).size;
}

public static boolean getSwapped (int jobID)
{
    return table.get(jobID - 1).swapped;
}

public static int getTimeLeft (int jobID)
{
    if (jobID != -1) {
        return (table.get(jobID - 1).maxCPUTime -
            table.get(jobID - 1).currentCPUTime);
    }
    else {
        return -1;
    }
}

public static int incrementIO(int jobID)
{
```

```
        Job incJob = table.get(jobID - 1);
        incJob.pendingIO++;
        // System.out.println("-JobTable increments I/O");
        // System.out.println("--Job# " + incJob.idNum + " has " + incJob.pendingIO + " i/o
requests");
        return incJob.pendingIO;
    }

    public static void incrementTime (int jobID, int time)
    {
        table.get(jobID - 1).currentCPUTime =
        table.get(jobID - 1).currentCPUTime + time;
    }

    public static void inMemory(int jobID)
    {
        table.get(jobID -1).inMemory = true;
    }

    public static boolean isBlocked(int jobID)
    {
        if (jobID != -1) {
            return table.get(jobID - 1).blocked;
        }
        else {
            return false;
        }
    }

    public static boolean isReady(int jobID)
    {
        if (jobID != -1) {
            return table.get(jobID - 1).ready;
        }
        else {
            return false;
        }
    }

    public static boolean isSwapping (int jobID)
    {
        return table.get(jobID-1).inDrum;
    }

    public static boolean isTerminated (int jobID)
    {
        return table.get(jobID-1).terminated;
    }

    public static void outMemory(int jobID)
    {
        table.get(jobID -1).inMemory = false;
    }

    public void print ()
    {
        // System.out.println("-JobTable Report");
```

```
    // System.out.print("--Jobs ");
    // for (int i = 0; i < table.size(); i++)
    // {
    //  String t = "";
    //  String b = "";
    //  String r = "";
    //  String io = ("(" + table.get(i).pendingIO + ")");
    //  if (table.get(i).terminated)
    //  {
    //      t = "T";
    //  }
    //  if (table.get(i).blocked)
    //  {
    //      b = "B";
    //  }
    //  if (table.get(i).ready)
    //  {
    //      r = "R";
    //  }
    //  System.out.print((table.get(i).idNum) +
    //      ":" + t + b + r + io + ", ");
    // }
    // System.out.println("");
}

public static void resetPriorityTime (int jobID)
{
    table.get(jobID -1).priorityTime = os.currentTime;
}

public static Job returnJob (int jobID)
{
    if (table.get(jobID - 1) != null) {
        return table.get(jobID - 1);
    }
    else {
        return null;
    }
}

public static void setAddress (int jobID, int address)
{
    table.get(jobID - 1).address = address;
}

public static void setBlocked (int jobID)
{
    if (jobID != -1) {
        // System.out.println("-JobTable sets " + jobID + " to blocked");
        table.get(jobID - 1).blocked = true;
    }
}

public static void setDirection (int jobID, int direction)
{
    table.get(jobID - 1).direction = direction;
    // System.out.println("-JobTable sets swap direction");
```

```
}

public static void setDoingIO (int jobID)
{
    if (jobID != -1) {
        // System.out.println("-JobTable sets " + jobID + " to latched");
        table.get(jobID - 1).latched = true;
    }
}

public static void setReady (int jobID)
{
    if (jobID != -1) {
        // System.out.println("-JobTable sets " + jobID + " as ready");
        table.get(jobID - 1).ready = true;
    }
}

public static void setSwapped (int jobID)
{
    table.get(jobID - 1).swapped = true;
}

public static void setSwapping (int jobID)
{
    table.get(jobID - 1).inDrum = true;
}

public static void stopSwapping (int jobID)
{
    table.get(jobID-1).inDrum = false;
}

public static void terminate(int jobID)
{
    table.get(jobID-1).terminated = true;
}

public static void unsetBlocked (int jobID)
{
    if (jobID != -1) {
        // System.out.println("-JobTable sets " + jobID + " to unblocked");
        table.get(jobID - 1).blocked = false;
    }
}

public static void unsetDoingIO (int jobID)
{
    if (jobID != -1) {
        // System.out.println("-JobTable sets " + jobID + " to unlatched");
        table.get(jobID - 1).latched = false;
    }
}

public static void unsetReady (int jobID)
{
    if (jobID != -1) {
```

```
            // System.out.println("-JobTable sets " + jobID + " to unready");
            table.get(jobID - 1).ready = false;
        }
    }
}
```