package cpu;

import Items.Cell;

import Items.Job;

import java.util.ArrayList;

/\*\*

\*

\* Responsible for viewing the GanttChart queue in the GUI frame by setting

\* the place and the color of every job representation

\*/

public class GanttChart {

// gantt chart

private static int ganttX =20 ; // start drawing point on x-coordinate

private static final int ganttY = 411; // Gantt location on y-coordinate

private static int ganttLastJob = 0; // show the number of the last job got represented in the Gantt chart

public static ArrayList<Cell> List = new ArrayList<Cell>(100); // list of gantt chart jobs' represnation

/\*\*

\* update the gantt chart representation by adding a new job cell

\* to the end.

\* @param job job to be represented in the gantt chart

\* @param time time of the simulation

\*/

public static void addJob(Job job , int time){

Cell cell ;

if(job == null)

{cell = Cell.createEmptyJobCell(ganttX, ganttY);} // represent empty job( white small cell)

else

{

if(job.jobNumber != ganttLastJob) // put 2 pixel margin between every two different jobs

{

ganttX += 1;

ganttLastJob = job.jobNumber;

}

cell = Cell.createGanttCell(ganttX, ganttY, job.jobNumber);

}

ganttX += 6; // set next job location

List.add(cell); // add cell to gantt chart list

if( (time+1) % 10 == 0 )

{

List.add(Cell.createMark(ganttX -1 , ganttY+50)); // put small black mark every 10 times

}

}

/\*\*

\* clear gantt chart list out of cell objects, and reset other variables

\*/

public static void clear(){

List.clear();

ganttX = 20; // start location on x-coordinate

ganttLastJob = 0; // default job number

}

}