Rotate an image by 90, 180 or 270 degrees using Java

The way the algorithm solves this problem is transforming the RGB image, pixel by pixel, into an array of integer elements. The matrix is very easy to rotate, in place through mathematical algorithms – multiplication with rotation matrices – or out of place by copying it into another matrix, in the appropriate order. I use the second method because the space needed to generate the rotation matrices exceeds the space needed to copy the original matrix. The rotation is done exclusively counter-clockwise.

The application is multithreaded, using the Producer/Consumer model. The Producer thread converts the RGB image information into an int array format, and the Consumer thread uses this information to build the image array. Threads enter NotRunnable after processing or loading a quarter of the image. The image is processed in horizontal strips. In order to know with certainty from which coordinate to start and where to stop processing at each stage of execution, the *controller* vector stores and transmits these coordinates, calculated before the execution of the threads.

The input images will be placed in the "input" directory, while the results of the processing will be stored in the "output" directory. The user will send in the command line the names of the 2 files - input and output, and the program will automatically find the indicated paths. The application processes only one image per run. The application receives as input an image in BITMAP 24 BIT RGB format and writes the output in the same format.

To comply with the 3-level inheritance requirement, these layers are represented by the degrees the image is rotated: 90 degree rotation will be level 1, 180 degree rotation will be level 2, 270 degree rotation will be level 3.

During runtime, the user will be prompted to select, from the console, by how many degrees they want the image to be rotated. If they enter an unrecognized value (allowed values are 90, 180, and 270), they will be prompted to enter a new, accepted value. At this point the user may cancel the

execution of the process by entering the value 0. The program will not write any new image, so the output directory will remain unchanged.

The user will be notified, in the console, what the statee of the processing image is in the Producer and Consumer threads.

To signal to the user that the execution has finished successfully, the execution duration of each step will be displayed in the console.