**Gebze Technical University**

**Computer Engineering**

**CSE 222 - 2018 Spring**

**HOMEWORK 5 REPORT**

**Melih Yabaş**

**161044072**

Course Assistant: Özgü Göksu

# INTRODUCTION

## Problem Definition

There is an image png or jpg format. We need to take its pixels and keep them in a priority queue. There are 3 different comparison method to do this. Bitmix, Lexicographical and Euclidean Norm. Our purpose is taking its pixels, adding and removing them in a multithreating program. We need 3 different comparator class, 4 threads, 1 pixel class, 1 priority queue and a main class. Thread 1 adds pixels to the queues. Thread 2, Thread 3 and Thread 4 deletes the top element of these queue. We had 3 priority queue object for 3 comparator. Each Thread (except Thread 1) cares only 1 queue. Comparator classes overrides their own compareTo methods. Pixel Class keeps red, green and blue values of pixels. Main class of the program creates Threads and tests all the methods.

## System Requirements

This program needs Java Virtual Machine to work properly.

Java Virtual Machine Requirements:

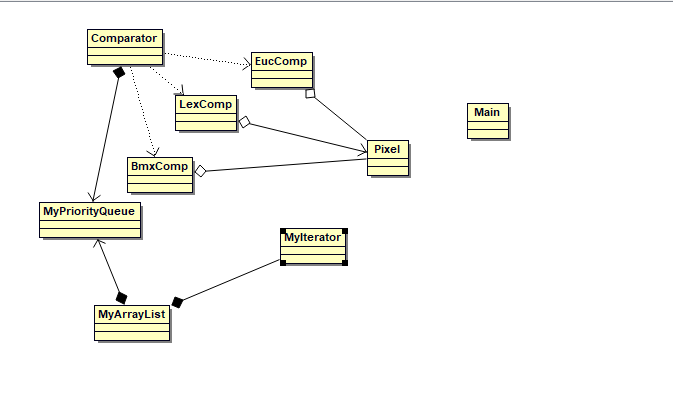
* Windows 10/8/7/Vista/XP/2000
* Windows Server 2008/2003
* Intel and 100% compatible processors are supported.
* Pentium 166 MHz or faster processor wit at least 64 MB of physical RAM.
* 98 MB for free disk space.

Packes:

* java.awt.\*
* java.awt.image.BufferedImage
* java.io
* javax.imageio.ImageIO
* java.util.Comparator
* java.util.Iterator

# METHOD

## Class Diagrams



## Problem Solution Approach

## There are 2 data structures in my program. One of them is Priority Queue. The other one is is an ArrayList. ArrayList keeps the elements of priority queue. We need to add and delete of element the priority queue. We have 3 object. Our threads take this job. Threads work together. So we need to synchronize them. We use offer and poll method in the threads. Therefor, these methods must be synchronized. Pixels that we added and polled keeped in a class. Thread 1 adds the elements and starts the other threads.

## Let say all the pixels number is n. Thread 1 rotates n times and runs the other ones. The other threads rotates n times too.

## Time complexity of the program:

## We creates object as a constant number. Space complexity of the program:

# RESULT

## Test Cases

An example image of to use have more than 300000 pixels. So, we cannot control it if is it correct we use all of them. I checked it for 400 pixels and I saw that the result is true. I control it more than 10 times and there was no error or warning. As the result screen, Thread 1 adds and element and the other Threads remove it. That is what I want. So program works certainly true.

## Running Results

