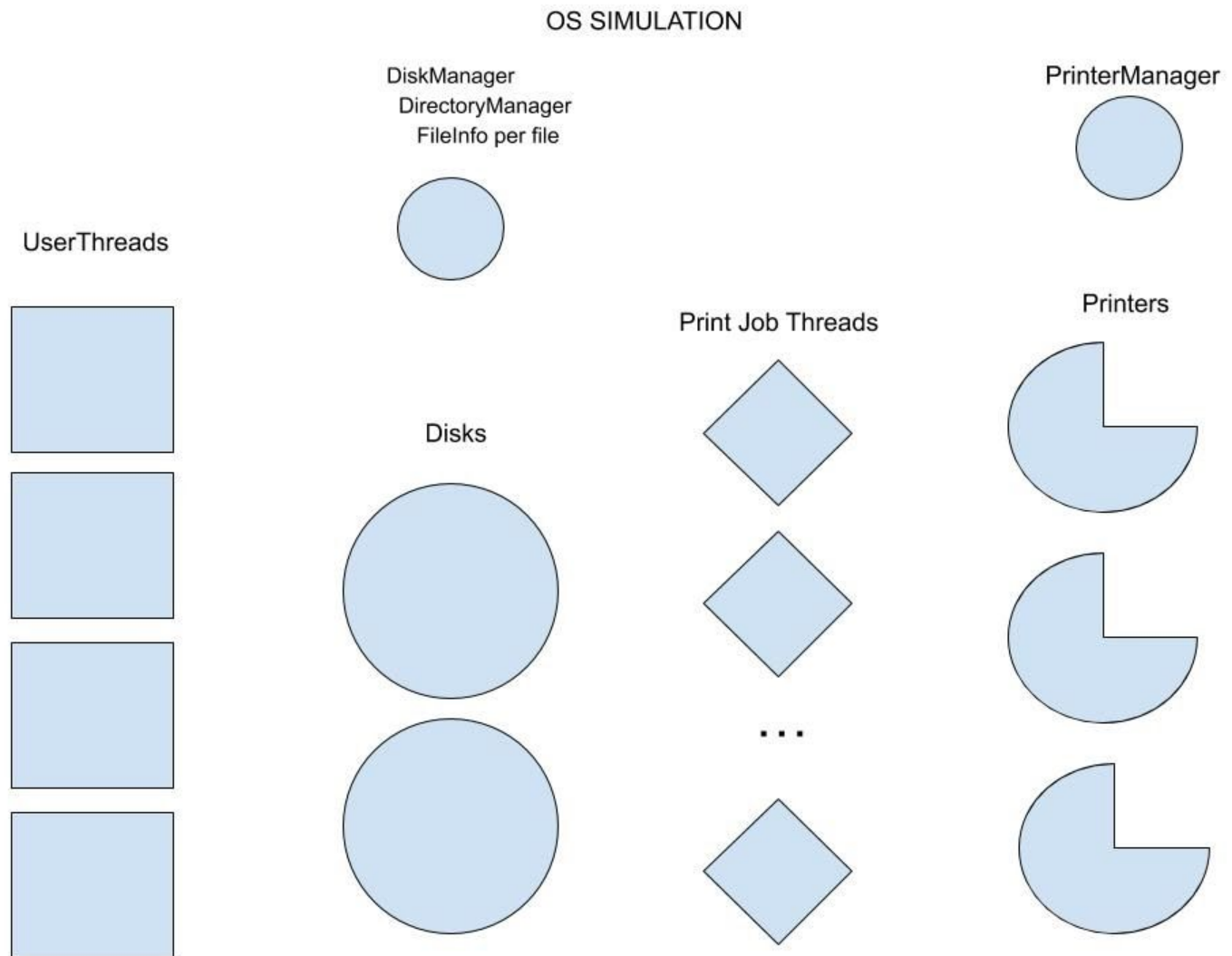


MAJOR SPOILER - DO NOT READ IF YOU WANT ANY CHALLENGE IN THIS ASSIGNMENT



The OS constructor can initialize users, disks, printers, diskManager, printerManager. The Disk and Printer essentially copy the data from the supplied StringBuffer into the appropriate place (disk sector or printer output file). I will elaborate only on UserThread and PrintJobThread. This is all off the top of my head, so you will still need to do some reading to get the Java stuff to work correctly. There are some tricky things about accessing the OS objects across methods, but there are several solutions that all work, so I will leave that up to you.

```

defun copy_StringBuffer(to, from)
  for each element of from:
    to[i] = from[i]

defun UserThread
  inputFile = open(file containing my user commands)
  processesCommandsIn(input file)
  close(inputFile)

defun processCommandsIn(inputFile)
  for each line in the file:
    switch (commandOf(line))
      ".save"
        saveFile(argumentOf(line))
      ".print"
        printFile(argumentOf(line)) // should be a one-liner, start a new PrintJobThread
    default:
      error("Unknown command:" line)

defun saveFile(name, inputFile)
  DiskNumber d = DiskManager.request()
  offset = DiskManager.getNextFreeSectorOnDisk(d)
  fileLines = 0
  for each line in the file: (note, we will stop when we see ".end")
    if line == ".end"
      DirectoryManager.enter(name, makeFileInfo(d, offset, fileLines))
      break
    else
      disks[d].write(offset + fileLines, line)
      fileLines++
  DiskManager.setNextFreeSectorOnDisk(d, offset+fileLines)
  DiskManager.release(d)

defun PrintJobThread(fileToPrint)
  StringBuffer line = new StringBuffer();
  FileInfo f = DirectoryManager.lookup(fileToPrint) // could there be an error here if f is undefined
  start = f.startingSector
  d = f.diskNumber
  PrinterNumber p = PrinterManager.request()
  for i in 0 to f.fileLength-1:
    disks[d].read(start+i, line) // note we can have many file readers, so no disk request required
    printers[p].print(line)
  PrinterManager.release(p)

```