Assignment 2

Practical Tasks in Advanced Java Part 2 (15%)

DUE DATE

Session 14

OBJECTIVE

- Apply fundamental programming concepts to design a program.
- Develop the program logic.
- Employ JavaFX concepts learned in the sessions so far.
- Manipulate and access files in Java.
- Perform search and sort operations in Java.
- Use XML serialization.
- Perform multiple transactions on the content of a file.

DESCRIPTION

For each of the following problems, start by developing the program logic, (Flowchart, or algorithm [pseudocode]), then develop each solution in Java code.

INSTRUCTIONS

1. Eight Queens

A puzzler for chess buffs is the Eight Queens problem, which asks: Is it possible to place eight queens on an empty chessboard so that no queen is “attacking” any other (i.e. no two queens are in the same row, in the same column, or along the same diagonal)? For example, if a queen is placed in the upper-left corner of the board, no other queens could be placed in any of the marked squares shown in Fig. 1 below.

Solve the problem recursively. [Hint: Your solution should begin with the first column and look for a location in that column where a queen can be placed — initially, place the queen in the first row. The solution should then recursively search the remaining columns. In the first few columns, there will be several locations where a queen may be placed. Take the first available location. If a column is reached with no possible location for a queen, the program should return to the previous column and move the queen in that column to a new row. This continuous backing up and trying new alternatives is an example of recursive backtracking.]

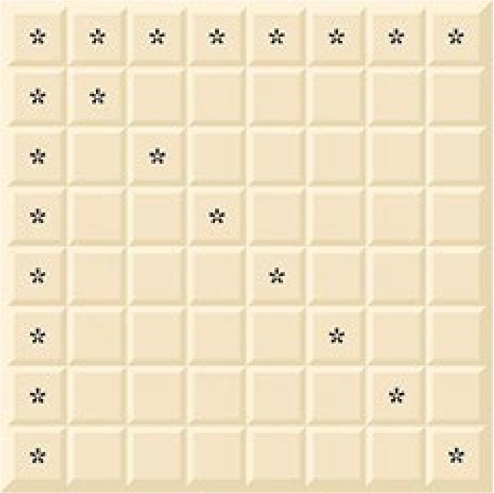

Fig.1 Squares eliminated by placing a queen in the upper-left corner of a chessboard.

2. Recursive File and Directory Manipulation

Using the String-processing capabilities in Chapter 14: “Strings, Characters and Regular Expressions,” the file and directory capabilities in “15.3 Using NIO Classes and Interfaces to Get File and Directory Information,” and “16.10 Maps” in the *Java How to Program, Early Objects* (11th Ed.) e-book, create an application that recursively walks a directory structure supplied by the user and reports the number of files of each file type (such as .java, .txt, .class, .docx, etc.) that exist in the specified directory path. Display the filename extensions in sorted order. Next, investigate method walk of class the Files. This method returns a stream that walks a directory and its subdirectories and returns the contents to you as a stream. Then reimplement the first part of this exercise, using lambdas and streams, rather than recursion.

3. Calculating Factorials with Lambdas and Streams

Reimplement the factorial methods of Figs. 18.3 and 18.4 in Chapter 18: “Recursion” in the e-book to calculate factorials using lambdas and streams rather than recursion.

SUBMISSION INSTRUCTIONS

Your submission must include a document with the logic for each problem in this assignment as well as each problem's source code.
Work must be submitted in the correct file type and be properly labelled as per the College naming convention:

NAME_COURSE_ASSIGNMENT. E.g. XuXiaLing_FM50D_A01.

GRADING CRITERIA

Assignment Value: **15%**

Grading Criteria	Grading
Eight Queens	/50
Recursive File and Directory Manipulation	/35
Calculating Factorials with Lambdas and Streams	/15
TOTAL	**/100**