2024 Java Programming Final Exam (40 points)

- Two points will be deducted for each item if the implementation differs from the suggestion made in the question or if an exception occurs while the program is running.
- There are no partial scores for each function (if there is only source code for the function and it does not work, no points are given)
- The same as other criteria that reduce an assignment's grade, such as compilation errors, comments(student name, student id), UTF-8 etc.
- Please submit only Java source files.
- It is not permitted to use AI tools such as ChatGPT, Copilot, etc.

1. Program for case-insensitive word search in a file(java.txt) (10 points)

submission: final1.zip

Implement a program that reads the given file(java.txt) line by line, stores it in a List (Vector, ArrayList, LinkedList), and then inputs a search word and outputs the corresponding line numbers and lines that matches the search word. (There is no limit on the class name and number of class files, but 0 points are given if List is not used.)

■ Reading java.txt and saving it to a List<String> (3점)

✓ Read the provided file line by line, add line numbers, and save the contents into the list.

■ Word search and search result output t (5점)

✓ Enter a search string on the screen and search for that string in the list.

✓ If there is a search string, both the line numbers and the line strings are displayed.

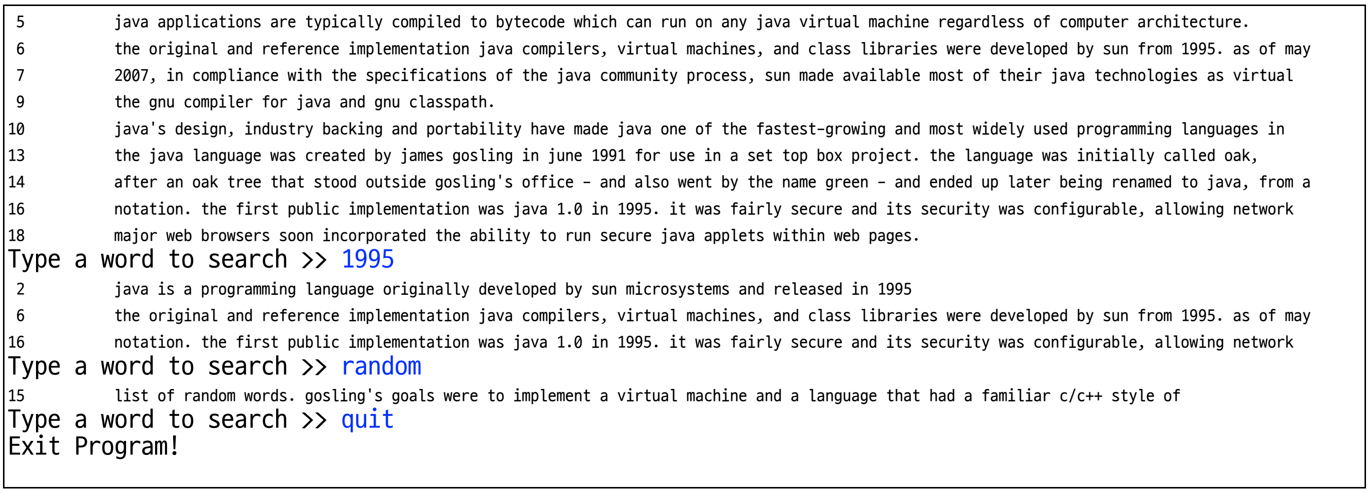
  - Case-insensitive search and print

✓ If there is no search string in the list, nothing is displayed.

✓ Enter "quit" to end the program. (2점)

■ Execution result

```
Type a word to search >> AAA          ←- "AAA" is not searched, nothing is displayed.
Type a word to search >> java
 1        java wiki
 2        java is a programming language originally developed by sun microsystems and released in 1995
 3        as a core component of sun's java platform. the language derives much of its syntax from c and c++
 5        java applications are typically compiled to bytecode which can run on any java virtual machine regardless of computer architecture.
 6        the original and reference implementation java compilers, virtual machines, and class libraries were developed by sun from 1995. as of may
 7        2007, in compliance with the specifications of the java community process, sun made available most of their java technologies as virtual
 9        the gnu compiler for java and gnu classpath.
10        java's design, industry backing and portability have made java one of the fastest-growing and most widely used programming languages in
13        the java language was created by james gosling in june 1991 for use in a set top box project. the language was initially called oak,
14        after an oak tree that stood outside gosling's office - and also went by the name green - and ended up later being renamed to java, from a
16        notation. the first public implementation was java 1.0 in 1995. it was fairly secure and its security was configurable, allowing network
18        major web browsers soon incorporated the ability to run secure java applets within web pages.
Type a word to search >> Java
 1        java wiki
 2        java is a programming language originally developed by sun microsystems and released in 1995
 3        as a core component of sun's java platform. the language derives much of its syntax from c and c++
```



```
 5          java applications are typically compiled to bytecode which can run on any java virtual machine regardless of computer architecture.
 6          the original and reference implementation java compilers, virtual machines, and class libraries were developed by sun from 1995. as of may
 7          2007, in compliance with the specifications of the java community process, sun made available most of their java technologies as virtual
 9          the gnu compiler for java and gnu classpath.
10          java's design, industry backing and portability have made java one of the fastest-growing and most widely used programming languages in
13          the java language was created by james gosling in june 1991 for use in a set top box project. the language was initially called oak,
14          after an oak tree that stood outside gosling's office - and also went by the name green - and ended up later being renamed to java, from a
16          notation. the first public implementation was java 1.0 in 1995. it was fairly secure and its security was configurable, allowing network
18          major web browsers soon incorporated the ability to run secure java applets within web pages.
Type a word to search >> 1995
 2          java is a programming language originally developed by sun microsystems and released in 1995
 6          the original and reference implementation java compilers, virtual machines, and class libraries were developed by sun from 1995. as of may
16          notation. the first public implementation was java 1.0 in 1995. it was fairly secure and its security was configurable, allowing network
Type a word to search >> random
15          list of random words. gosling's goals were to implement a virtual machine and a language that had a familiar c/c++ style of
Type a word to search >> quit
Exit Program!
```

2. KTX reservation system using multithread (30 points)

submission: final2.zip

Implement a system in which five people(thread) can simultaneously reserve 40 KTX seats. (If thread execution and shared resource synchronization are not used, 0 points are given.)

- ■ Implementation files: KexReservatopmApp.java, class file for a shared resource, thread class, etc. (No limit)
- ■ Operation process
  - ✓ Five Person threads randomly select a seat between rows (0 to 3), columns (0 to 9), and display the seats in their own colors if seats are not selected by other threads.
  - ✓ Thread termination
    - – If there are no empty seats, all threads terminate.
    - – When all threads terminate, the reservation information is saved to a file ("reservation.out")
    - – Each thread can save the reservation information as a file during the reservation process.
- ■ GUI operation: KtxRervationApp.java
  - ✓ Initial operation (provides basic source codes)
    - – Creation of 40 JLabels and placement on screen
    - – Placement of 2 JButtons

- ■ **Implementation of "Start"** button functionality: start reservation button **(20 points)**
  - ✓ ActionListener implementation of "Start" button (2 points)
  - ✓ Creation and operation of five Person threads (4 points)
    - – Thread name: P1, P2, P3, P4, P5
    - – Display color information for each thread
      - ➢ P1: Color.orange, P2: Color.yellow, P3: Color.cyan,
      - ➢ P4: Color.magenta, P5: Color.pink
    - – Person Threads operate at sleep(1000) intervals.
    - – If there are no empty seats to select, all threads are terminated.
  - ✓ Seat reservation (8 points)
    - – Each thread randomly generates row and column numbers and select a seat
    - – When an empty seat is selected, the seat number and thread name are displayed in JLabel. (3 points)
    - – The contents below are displayed on the screen (console window). (3 points)
      - ➢ Console output: "[Reservation Success] P1: seatnumber, Remained seats: remained seat number "

➢ The line break character is implemented as follows: use the <br> tag

```
String seatString = Integer.toString(col+1) + rowString[row];
String htmlText = String.format("<html><center> %s <br> %s </center></html>",
                               seatString, name);
seatLabels[i][j].setText(htmlText);
```

– The contents below are printed and skipped if the seat chosen by one thread has already been chosen by another thread. (2 points)

➢ Console output: “[Reservation Fail] P5: 5B was already reserved.”

➢ No display on JLabel

✓ **Storing reserved seat information for Replay function** (4 points)

– Save the reservation information to a file only when the reservation is successful by selecting an empty seat.

➢ File name: “reservation.out”

➢ No restrictions on file format (text, binary or object file)

➢ File contents: seat row, col, thread name (can be configured as separate objects)

■ **Implementation of “Replay” button function**: view reservation status again (10 points)

✓ ActionListener implementation of “Replay” button (2 points)

✓ Implementation of actions (8 points)

– Reset the thread name and color displayed in JLabel (2 points)

– Read the “reservation.out” file and display it on the screen again at 500ms intervals

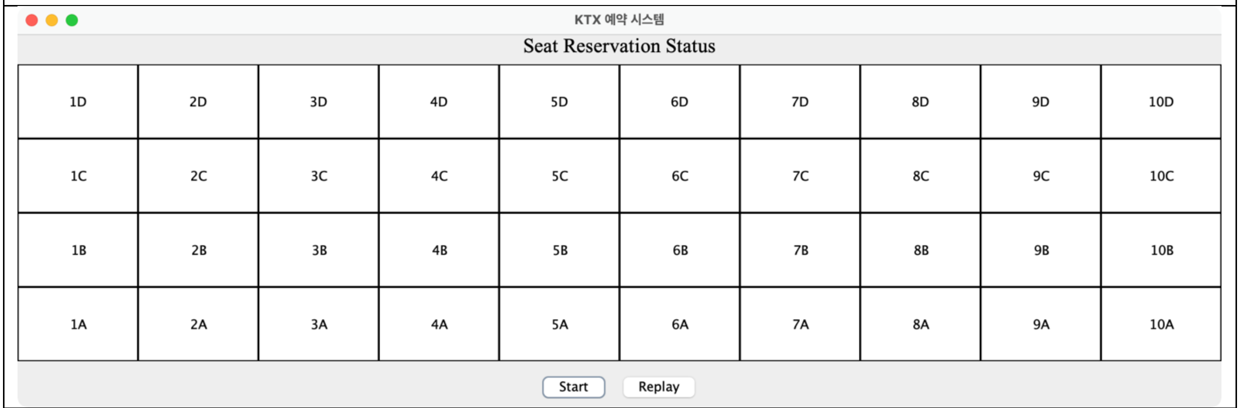
➢ Display the color and name of each thread as in the normal reservation process (6 points)

– A separate thread can be created for the replay function. (optional)

– Please refer to the video for detailed operation process.

■ **GUI screen**

Initial screen

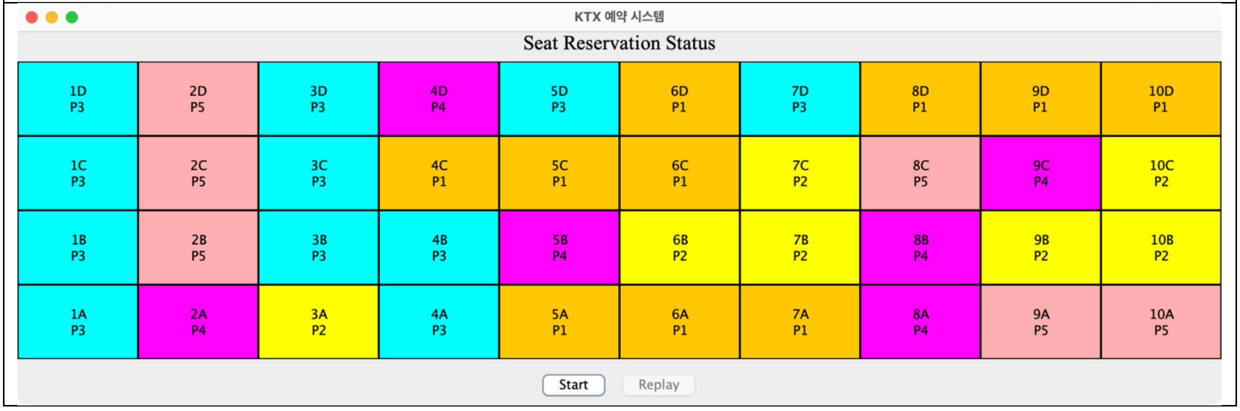


KTX 예약 시스템

Seat Reservation Status

1D	2D	3D	4D	5D	6D	7D	8D	9D	10D
1C	2C	3C	4C	5C	6C	7C	8C	9C	10C
1B	2B	3B	4B	5B	6B	7B	8B	9B	10B
1A	2A	3A	4A	5A	6A	7A	8A	9A	10A

Reservation end screen



Seat Reservation Status

1D P3	2D P5	3D P3	4D P4	5D P3	6D P1	7D P3	8D P1	9D P1	10D P1
1C P3	2C P5	3C P3	4C P1	5C P1	6C P1	7C P2	8C P5	9C P4	10C P2
1B P3	2B P5	3B P3	4B P3	5B P4	6B P2	7B P2	8B P4	9B P2	10B P2
1A P3	2A P4	3A P2	4A P3	5A P1	6A P1	7A P1	8A P4	9A P5	10A P5

- ■ **Execution result**

```
[Reservation Success] P1:  4C, Remained seats: 39
[Reservation Success] P5:  9A, Remained seats: 38
[Reservation Success] P4:  5B, Remained seats: 37
[Reservation Success] P3:  7D, Remained seats: 36
[Reservation Success] P2:  7B, Remained seats: 35
[Reservation Success] P1:  6A, Remained seats: 34
[Reservation Success] P5: 10A, Remained seats: 33
[Reservation Success] P4:  8B, Remained seats: 32
[Reservation Success] P3:  1D, Remained seats: 31
[Reservation Success] P2:  7C, Remained seats: 30
[Reservation Success] P5:  8C, Remained seats: 29
[Reservation Fail] P3:  5B was already reserved.
[Reservation Fail] P4:  5B was already reserved.
[Reservation Fail] P1:  7D was already reserved.
[Reservation Success] P2: 10C, Remained seats: 28
[Reservation Fail] P5:  5B was already reserved.

.  .  . skip intermediate steps

[Reservation Fail] P1:  7D was already reserved.
[Reservation Fail] P3: 10B was already reserved.
[Reservation Fail] P2: 10C was already reserved.
[Reservation Success] P5:  2C, Remained seats:  1
[Reservation Fail] P3:  6B was already reserved.
[Reservation Fail] P2:  3C was already reserved.
[Reservation Fail] P4:  5A was already reserved.
[Reservation Success] P1:  7A, Remained seats:  0
Person Thread: P5 Terminated!
Person Thread: P3 Terminated!
Person Thread: P2 Terminated!
Person Thread: P4 Terminated!
Person Thread: P1 Terminated!
All Threads Terminated
Write reservation.out
```

```
Intialize reservation information
Read reservation.out
[1][3]: P1
[3][8]: P5
[2][4]: P4
[0][6]: P3

.  .  . skip intermediate steps

[0][7]: P1
[1][8]: P4
[0][8]: P1
[2][3]: P3
[1][1]: P5
[3][6]: P1
Replay is done!
```