/\*\*

\*The task is similar to the producer–consumer problem discussed in the video.

\*The QuestionBuffer class is used by two threads.

\*The TeacherThread writes some questions to the QuestionBuffer at a random interval.

\*The StudentThread reads the Buffer whenever there are new values and prints those questions.

\*If the Buffer is empty, the StudentThread goes to a suspended state.

\*Whenever a new question is inserted, the TeacherThread gives the suspended StudentThread a wake-up call.

\*You only need to modify the QuestionBuffer class and print the buffer pointer at the end of the main method.

\* Best of luck.

\*/

import java.util.ArrayList;

import java.util.List;

class QuestionBuffer {

public int pointer = -1;

private List<String> registers = new ArrayList<>();

public synchronized String readQuestionFromReg() throws InterruptedException {

/\*

\* TODO:

\* 1. Use monitor for this method [public synchronized String readQuestionFromReg()].

\* 2. If the pointer is at -1, it means there is no item in the list. So, suspend the StudentThread [using wait()].

\* 3. Take the value from the the list.

\* 4. Decrease the pointer by 1.

\* 5. Remove the string from the list,

\* 6. Return the value

\*/

if(pointer == -1) {

wait();

}

pointer--;

return registers.remove(0);

}

public synchronized void writeQuestionToReg(String value) {

/\*\*

\* TODO:

\* 1. Use monitor for this method [public synchronized String writeQuestionToReg()].

\* 2. Add the string to the list.

\* 3. Increase the pointer by 1.

\* 4. If the pointer is at 0, it indicates first item in the list. So, wakeup StudentThread [using notifyAll()].

\*/

pointer++;

registers.add(value);

if(pointer == 0) {

notifyAll();

}

}

}

// Do not modify this class

class TeacherThread extends Thread {

private String[] values = {

"What is your name?",

"What is your student ID?",

"Please mention your course title,theory section and lab section.",

"Have you received your first dose of covid-19 vaccine?",

"Have you received your second dose of covid-19 vaccine?",

"Are you prepared for offline classes in the upcoming semester?",

"Write a few lines to describe yourself.",

};

QuestionBuffer questionBuffer;

public TeacherThread(QuestionBuffer questionBuffer) {

this.questionBuffer = questionBuffer;

}

@Override

public void run() {

for (int i = 0 ; i < values.length ; i++) {

try {

questionBuffer.writeQuestionToReg(values[i]);

sleep((int)(Math.random() \* 1000));

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

// Do not modify this class

class StudentThread extends Thread {

QuestionBuffer questionBuffer;

public StudentThread(QuestionBuffer questionBuffer) {

this.questionBuffer = questionBuffer;

}

@Override

public void run() {

try {

while (true) {

System.out.println(Thread.currentThread().getName() + " prints: " + questionBuffer.readQuestionFromReg());

}

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

// Do not modify this class

public class LabTask{

public static void main(String[] args) throws InterruptedException {

QuestionBuffer questionBuffer = new QuestionBuffer();

StudentThread studentThread = new StudentThread(questionBuffer);

TeacherThread teacherThread = new TeacherThread(questionBuffer);

teacherThread.start();

studentThread.start();

teacherThread.join();

studentThread.stop();

//TO-DO: Print the buffer pointer.

System.out.println("The buffer pointer is at: " + questionBuffer.pointer);

}

}