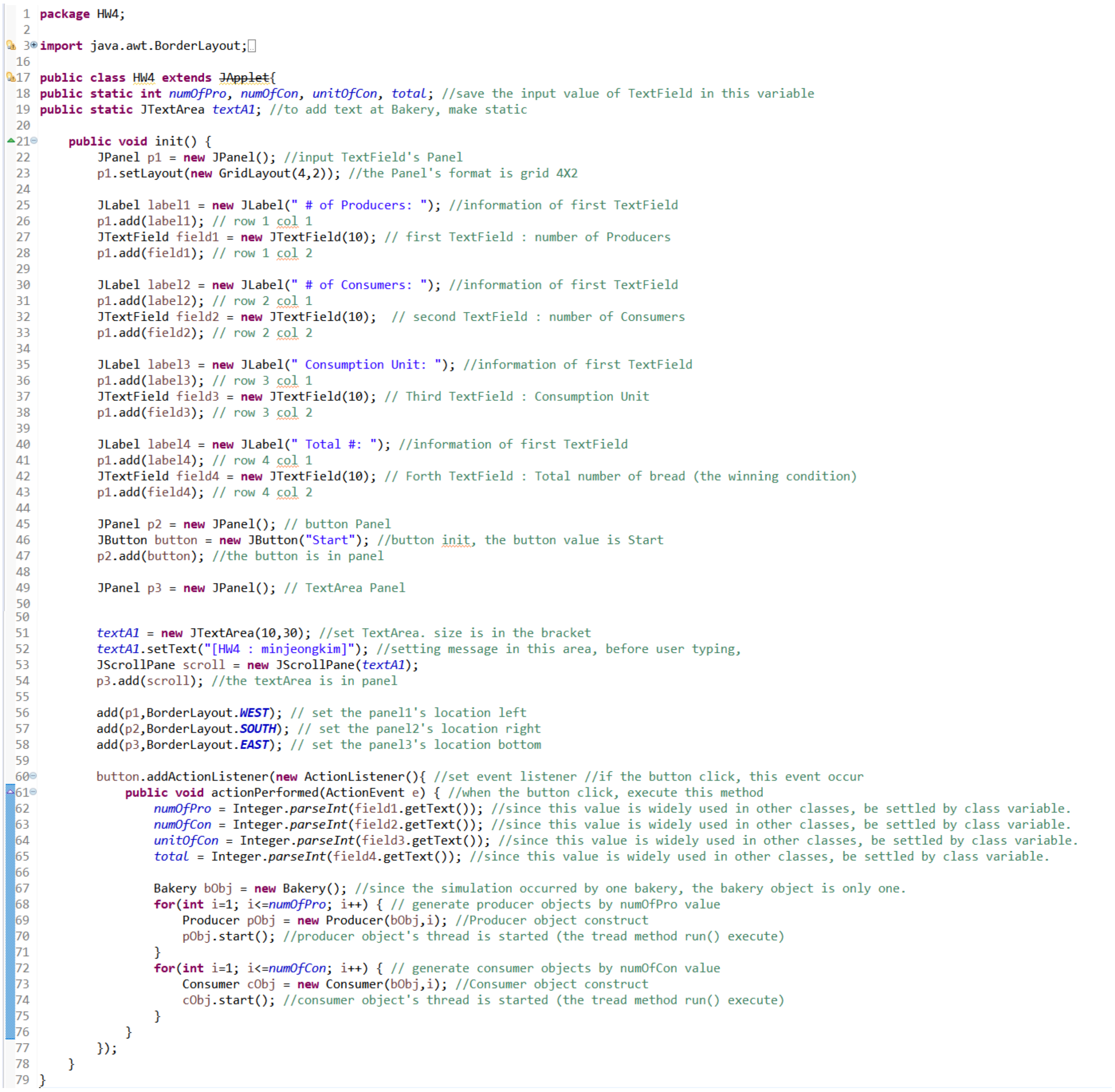
# REPORT

## JAVA Programming and Labs

**Student ID : 1771008**
**Name : Minjeong Kim**

## HW4.java : main method class

### [Code]



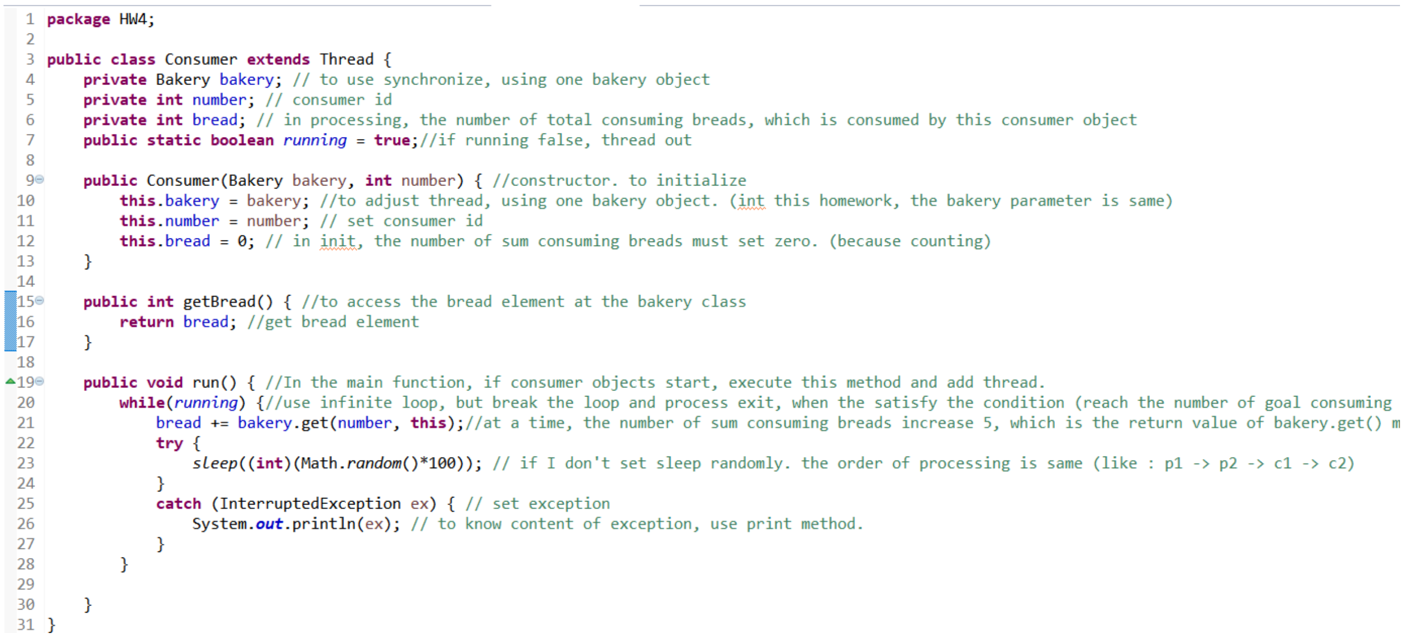
```java
package HW4;

import java.awt.BorderLayout;

public class HW4 extends JApplet{
public static int numOfPro, numOfCon, unitOfCon, total; //save the input value of TextField in this variable
public static JTextArea textA1; //to add text at Bakery, make static

    public void init() {
        JPanel p1 = new JPanel(); //input TextField's Panel
        p1.setLayout(new GridLayout(4,2)); //the Panel's format is grid 4X2

        JLabel label1 = new JLabel(" # of Producers: "); //information of first TextField
        p1.add(label1); // row 1 col 1
        JTextField field1 = new JTextField(10); // first TextField : number of Producers
        p1.add(field1); // row 1 col 2

        JLabel label2 = new JLabel(" # of Consumers: "); //information of first TextField
        p1.add(label2); // row 2 col 1
        JTextField field2 = new JTextField(10);  // second TextField : number of Consumers
        p1.add(field2); // row 2 col 2

        JLabel label3 = new JLabel(" Consumption Unit: "); //information of first TextField
        p1.add(label3); // row 3 col 1
        JTextField field3 = new JTextField(10); // Third TextField : Consumption Unit
        p1.add(field3); // row 3 col 2

        JLabel label4 = new JLabel(" Total #: "); //information of first TextField
        p1.add(label4); // row 4 col 1
        JTextField field4 = new JTextField(10); // Forth TextField : Total number of bread (the winning condition)
        p1.add(field4); // row 4 col 2

        JPanel p2 = new JPanel(); // button Panel
        JButton button = new JButton("Start"); //button init, the button value is Start
        p2.add(button); //the button is in panel

        JPanel p3 = new JPanel(); // TextArea Panel


        textA1 = new JTextArea(10,30); //set TextArea. size is in the bracket
        textA1.setText("[HW4 : minjeongkim]"); //setting message in this area, before user typing,
        JScrollPane scroll = new JScrollPane(textA1);
        p3.add(scroll); //the textArea is in panel

        add(p1,BorderLayout.WEST); // set the panel1's location left
        add(p2,BorderLayout.SOUTH); // set the panel2's location right
        add(p3,BorderLayout.EAST); // set the panel3's location bottom

        button.addActionListener(new ActionListener(){ //set event listener //if the button click, this event occur
            public void actionPerformed(ActionEvent e) { //when the button click, execute this method
                numOfPro = Integer.parseInt(field1.getText()); //since this value is widely used in other classes, be settled by class variable.
                numOfCon = Integer.parseInt(field2.getText()); //since this value is widely used in other classes, be settled by class variable.
                unitOfCon = Integer.parseInt(field3.getText()); //since this value is widely used in other classes, be settled by class variable.
                total = Integer.parseInt(field4.getText()); //since this value is widely used in other classes, be settled by class variable.

                Bakery bObj = new Bakery(); //since the simulation occurred by one bakery, the bakery object is only one.
                for(int i=1; i<=numOfPro; i++) { // generate producer objects by numOfPro value
                    Producer pObj = new Producer(bObj,i); //Producer object construct
                    pObj.start(); //producer object's thread is started (the tread method run() execute)
                }
                for(int i=1; i<=numOfCon; i++) { // generate consumer objects by numOfCon value
                    Consumer cObj = new Consumer(bObj,i); //Consumer object construct
                    cObj.start(); //consumer object's thread is started (the tread method run() execute)
                }
            }
        });
    }
}
```

### [Code analysis]

It is more efficient to compare all files with the previous homework HW3_2 package. I will explain the only exchange code. The biggest change is the removal of the frame variable and setting the applet. To set the applet, I use JApplet class by using extends. And remove or revise all the variable related with frame. However. it is similar to the code logic for creating the previous GUI.

## Consumer.java

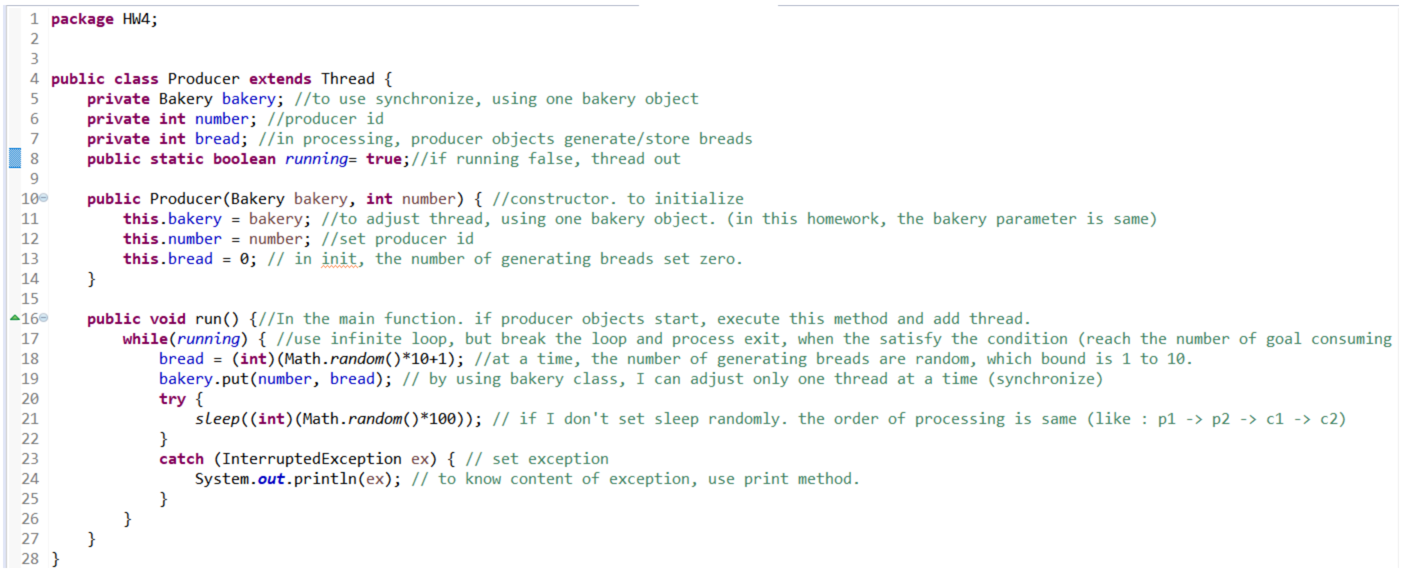
### [Code]



```java
package HW4;

public class Consumer extends Thread {
    private Bakery bakery; // to use synchronize, using one bakery object
    private int number; // consumer id
    private int bread; // in processing, the number of total consuming breads, which is consumed by this consumer object
    public static boolean running = true;//if running false, thread out

    public Consumer(Bakery bakery, int number) { //constructor. to initialize
        this.bakery = bakery; //to adjust thread, using one bakery object. (int this homework, the bakery parameter is same)
        this.number = number; // set consumer id
        this.bread = 0; // in init, the number of sum consuming breads must set zero. (because counting)
    }

    public int getBread() { //to access the bread element at the bakery class
        return bread; //get bread element
    }

    public void run() { //In the main function, if consumer objects start, execute this method and add thread.
        while(running) {//use infinite loop, but break the loop and process exit, when the satisfy the condition (reach the number of goal consuming
            bread += bakery.get(number, this);//at a time, the number of sum consuming breads increase 5, which is the return value of bakery.get() m
            try {
                sleep((int)(Math.random()*100)); // if I don't set sleep randomly. the order of processing is same (like : p1 -> p2 -> c1 -> c2)
            }
            catch (InterruptedException ex) { // set exception
                System.out.println(ex); // to know content of exception, use print method.
            }
        }

    }
}
```

## Consumer.java

### [Code]



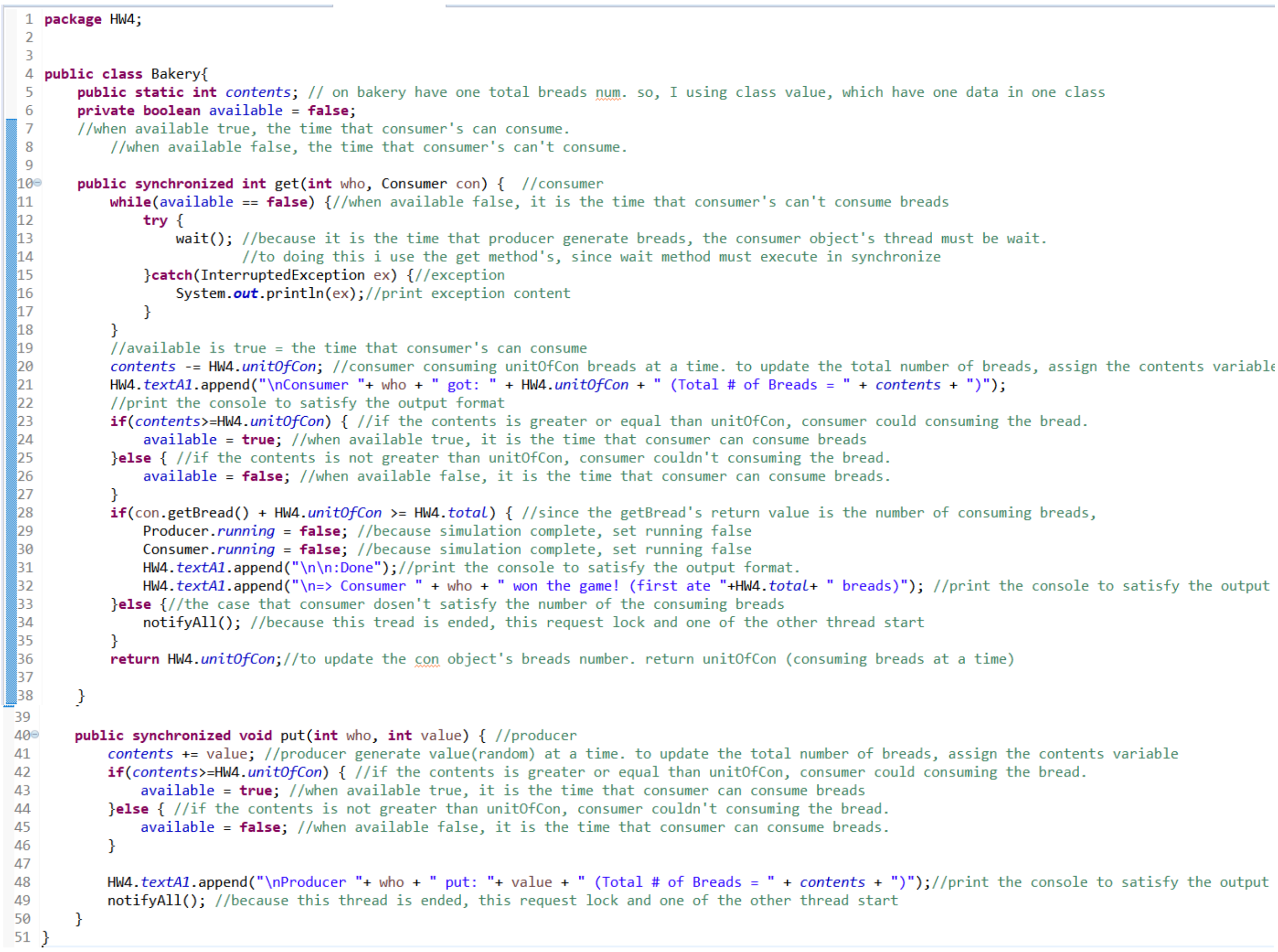
```java
package HW4;


public class Producer extends Thread {
    private Bakery bakery; //to use synchronize, using one bakery object
    private int number; //producer id
    private int bread; //in processing, producer objects generate/store breads
    public static boolean running= true;//if running false, thread out

    public Producer(Bakery bakery, int number) { //constructor. to initialize
        this.bakery = bakery; //to adjust thread, using one bakery object. (in this homework, the bakery parameter is same)
        this.number = number; //set producer id
        this.bread = 0; // in init, the number of generating breads set zero.
    }

    public void run() {//In the main function. if producer objects start, execute this method and add thread.
        while(running) { //use infinite loop, but break the loop and process exit, when the satisfy the condition (reach the number of goal consuming
            bread = (int)(Math.random()*10+1); //at a time, the number of generating breads are random, which bound is 1 to 10.
            bakery.put(number, bread); // by using bakery class, I can adjust only one thread at a time (synchronize)
            try {
                sleep((int)(Math.random()*100)); // if I don't set sleep randomly. the order of processing is same (like : p1 -> p2 -> c1 -> c2)
            }
            catch (InterruptedException ex) { // set exception
                System.out.println(ex); // to know content of exception, use print method.
            }
        }
    }
}
```

### [Code analysis] – Consumer, Producer

It have a small change, adding running variable. Previously, I didn’t put a condition to terminate the tread, but just used system.exit(0) if it satisfied the condition of the bakery running. Because, I just see the result in the console. Howerver in this case, I must see the result in applet. So, I set the thread terminal condition by using the static variable running. if the running is true, the thread looping, if not, the thread is terminated.

## Bakery.java

### [Code]



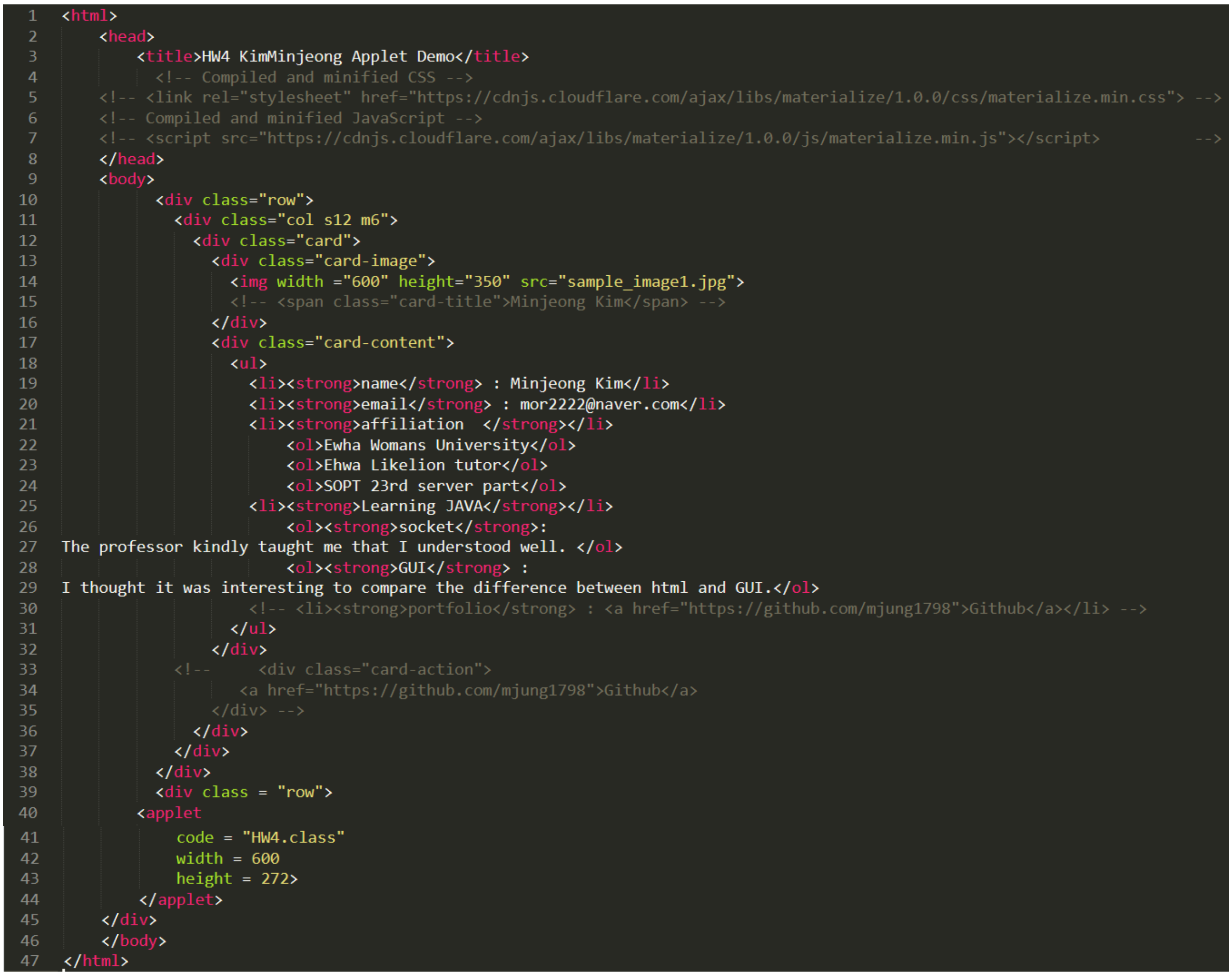
```java
package HW4;


public class Bakery{
    public static int contents; // on bakery have one total breads num. so, I using class value, which have one data in one class
    private boolean available = false;
    //when available true, the time that consumer's can consume.
        //when available false, the time that consumer's can't consume.

    public synchronized int get(int who, Consumer con) {  //consumer
        while(available == false) {//when available false, it is the time that consumer's can't consume breads
            try {
                wait(); //because it is the time that producer generate breads, the consumer object's thread must be wait.
                        //to doing this i use the get method's, since wait method must execute in synchronize
            }catch(InterruptedException ex) {//exception
                System.out.println(ex);//print exception content
            }
        }
        //available is true = the time that consumer's can consume
        contents -= HW4.unitOfCon; //consumer consuming unitOfCon breads at a time. to update the total number of breads, assign the contents variable
        HW4.textA1.append("\nConsumer "+ who + " got: " + HW4.unitOfCon + " (Total # of Breads = " + contents + ")");
        //print the console to satisfy the output format
        if(contents>=HW4.unitOfCon) { //if the contents is greater or equal than unitOfCon, consumer could consuming the bread.
            available = true; //when available true, it is the time that consumer can consume breads
        }else { //if the contents is not greater than unitOfCon, consumer couldn't consuming the bread.
            available = false; //when available false, it is the time that consumer can consume breads.
        }
        if(con.getBread() + HW4.unitOfCon >= HW4.total) { //since the getBread's return value is the number of consuming breads,
            Producer.running = false; //because simulation complete, set running false
            Consumer.running = false; //because simulation complete, set running false
            HW4.textA1.append("\n\n:Done");//print the console to satisfy the output format.
            HW4.textA1.append("\n=> Consumer " + who + " won the game! (first ate "+HW4.total+ " breads)"); //print the console to satisfy the output
        }else {//the case that consumer dosen't satisfy the number of the consuming breads
            notifyAll(); //because this tread is ended, this request lock and one of the other thread start
        }
        return HW4.unitOfCon;//to update the con object's breads number. return unitOfCon (consuming breads at a time)

    }

    public synchronized void put(int who, int value) { //producer
        contents += value; //producer generate value(random) at a time. to update the total number of breads, assign the contents variable
        if(contents>=HW4.unitOfCon) { //if the contents is greater or equal than unitOfCon, consumer could consuming the bread.
            available = true; //when available true, it is the time that consumer can consume breads
        }else { //if the contents is not greater than unitOfCon, consumer couldn't consuming the bread.
            available = false; //when available false, it is the time that consumer can consume breads.
        }

        HW4.textA1.append("\nProducer "+ who + " put: "+ value + " (Total # of Breads = " + contents + ")");//print the console to satisfy the output
        notifyAll(); //because this thread is ended, this request lock and one of the other thread start
    }
}
```

### [Code analysis]

Previously, I use the System.out.println() function, to show the result in the console. However, in this case, since I have to show the result in the applet textarea, I use the append method. By setting the textArea variable to class variable, I can use it in Bakery class. And, as I said before, the applet should not be terminated to show the result. therefore, I set the Producer and Consumer running variable set false, to terminate the thread.

## Index.java

### [Code]

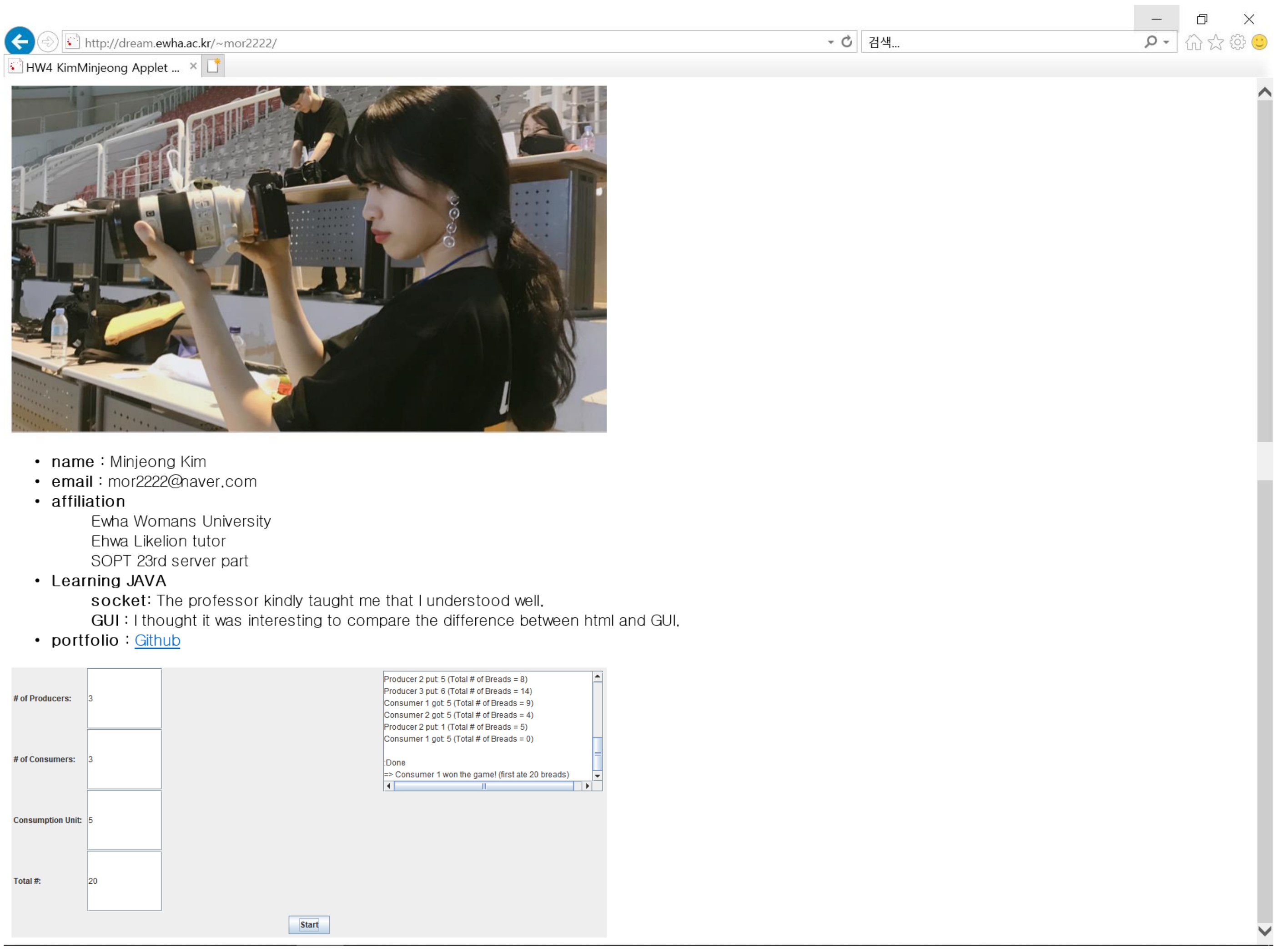


```html
<html>
    <head>
        <title>HW4 KimMinjeong Applet Demo</title>
          <!-- Compiled and minified CSS -->
    <!-- <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css"> -->
    <!-- Compiled and minified JavaScript -->
    <!-- <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>          -->
    </head>
    <body>
          <div class="row">
            <div class="col s12 m6">
              <div class="card">
                <div class="card-image">
                  <img width ="600" height="350" src="sample_image1.jpg">
                  <!-- <span class="card-title">Minjeong Kim</span> -->
                </div>
                <div class="card-content">
                  <ul>
                    <li><strong>name</strong> : Minjeong Kim</li>
                    <li><strong>email</strong> : mor2222@naver.com</li>
                    <li><strong>affiliation  </strong></li>
                        <ol>Ewha Womans University</ol>
                        <ol>Ehwa Likelion tutor</ol>
                        <ol>SOPT 23rd server part</ol>
                    <li><strong>Learning JAVA</strong></li>
                        <ol><strong>socket</strong>:
The professor kindly taught me that I understood well. </ol>
                        <ol><strong>GUI</strong> :
I thought it was interesting to compare the difference between html and GUI.</ol>
                    <!-- <li><strong>portfolio</strong> : <a href="https://github.com/mjung1798">Github</a></li> -->
                  </ul>
                </div>
            <!--     <div class="card-action">
                   <a href="https://github.com/mjung1798">Github</a>
                </div> -->
              </div>
            </div>
          </div>
          <div class = "row">
        <applet
            code = "HW4.class"
            width = 600
            height = 272>
        </applet>
    </div>
    </body>
</html>
```

### [Code analysis]

I have learned web development before. So, using various tags, I made it easy to see my information in html. According to the attribute of the tag, the child-parent relationship is properly arranged.

## [Output Website]



- name : Minjeong Kim
- email : mor2222@naver.com
- affiliation
    Ewha Womans University
    Ehwa Likelion tutor
    SOPT 23rd server part
- Learning JAVA
    socket: The professor kindly taught me that I understood well.
    GUI : I thought it was interesting to compare the difference between html and GUI.
- portfolio : Github

| | |
|---|---|
| # of Producers: | 3 |
| # of Consumers: | 3 |
| Consumption Unit: | 5 |
| Total #: | 20 |

```
Producer 2 put: 5 (Total # of Breads = 8)
Producer 3 put: 6 (Total # of Breads = 14)
Consumer 1 got: 5 (Total # of Breads = 9)
Consumer 2 got: 5 (Total # of Breads = 4)
Producer 2 put: 1 (Total # of Breads = 5)
Consumer 1 got: 5 (Total # of Breads = 0)

:Done
=> Consumer 1 won the game! (first ate 20 breads)
```

## What I learned from this homework 4

### <applet java version>

I have had a lot of trouble because of the problem that the applet does not run in my web browser. So, I changed the settings of the java control panel first, but it was different from the guidelines, so I didn't. But I tried more. searching the various tabs and finding the exception web site. It seemed to be resolved! However adversity remained. the internet explore didn't show anything related to Java. I tried many things and found problems.

The version of java 8, which the applet is applied, is java 1.0.8_191. However, my java 8 version is java 1.0.8_161. So, I update in this version the professor's page is completely loaded!!! So, I tried to view my page, but an error occurred. The compiled version and the read version did not match. My class files are compiled in java 1.0.8_161, however the class files for viewing the applet on the web should have been compiled in java 1.0.8_191.

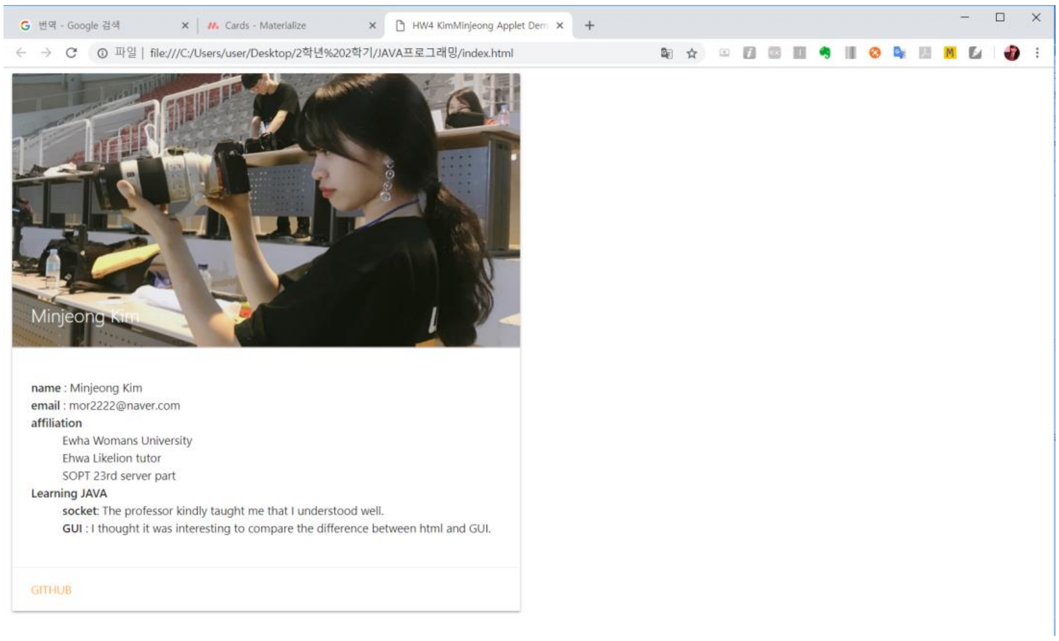
To compile java 1.0.8_191 version, I change the control, I set the environment variable again in control panel. And then, the working successfully

I have learned that the version of language is important to use any library. such as applet is not used in the HTML5.

### <library conflict - materialize>

At first I wanted to design nicely the html page that have information part, not the applet part. Although it was possible to use CSS and javascript, there were many inconvenient for remaining time. So I select the library materialize to design page conveniently.

So first I wanted to design page like bottom image


G 번역 - Google 검색
Cards - Materialize
HW4 KimMinjeong Applet Dem
파일 | file:///C:/Users/user/Desktop/2학년%202학기/JAVA프로그래밍/index.html
Minjeong Kim
GITHUB

**name** : Minjeong Kim
**email** : mor2222@naver.com
**affiliation**
Ewha Womans University
Ehwa Likelion tutor
SOPT 23rd server part
**Learning JAVA**
**socket**: The professor kindly taught me that I understood well.
**GUI** : I thought it was interesting to compare the difference between html and GUI.

However, if do, the applet part have scrolling trouble. because of the animation that material design having. So I didn't implement this design, and annotate.

**link is )** https://materializecss.com/