

Grupped medlemmer: Vladimirs Civilgins

NB! Hele besvarelsen er gjort i vscode!

Oppgave 1

The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project structure with folders 'Obligt' and 'Oppgave_1', 'Oppgave_2', and 'Oppgave_3'. The 'Obligt' folder contains 'Input.java', 'Main.java', and 'Output.java'. The 'Oppgave_1' folder contains 'Hamburger.java', 'HamburgerBrett.java', 'HamburgerSjappe.java', 'Kokk.java', 'Servitor.java', and 'Oblig1.iml'. The 'Oppgave_2' folder contains 'Hamburgersjappe.java', 'Kokk.java', and 'Servitor.java'. The 'Oppgave_3' folder contains 'Hamburgersjappe.java', 'Kokk.java', and 'Servitor.java'. The main editor shows the code for 'Input.java' in the 'src' folder. The code is as follows:

```
src > Oppgave_1 > Input.java > Input > run()
1 package Oppgave_1;
2
3 import javax.swing.JOptionPane;
4
5 public class Input implements Runnable{
6
7     @Override
8     public void run() {
9         while(!Main.melding.equals(anObject: "quit")){
10             Main.melding = JOptionPane.showInputDialog(message: "Input verdi (quit for å avslutte)");
11         }
12     }
13 }
14
15
```

Below the code editor, the TERMINAL pane shows the output of the program:

```
PS C:\Users\vlads\Desktop\HVL\Studie\DAT108\Obligt> & "C:\Program Files\Java\jdk-12.0.2\bin\java.exe" -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:59260' -cp 'C:\Users\vlads\AppData\Roaming\Code\User\workspaceStorage\1b6afb10b1744d08b68fbc486668033c\redhat.java Obligt_409d3d1c\bin' 'Oppgave_1.Main'
Hello, verden
Hello, verden
Hello, verden
Hello, verden
Hello, verden
Hello, verden
Hello, verden
Hello, verden
Hello, verden
Hei
Hei
Hei
Hei
Hei
```

Overlappende på terminalen er et dialogboks-fenster med tittelen 'Input'. Den inneholder teksten 'Input verdi (quit for å avslutte)' og et tekstfelt. Nederst i dialogboksen er det to knapper: 'OK' og 'Cancel'.

Oppgave 2

src > Oppgave_2 > HamburgerSjappe.java > ...

```
1 package Oppgave_2;
2
3 import java.util.Arrays;
4
5 public class HamburgerSjappe {
6
7     public static void main(String[] args) {
8
9         final String[] kokker = {"Anne", "Erik", "Knut"};
10        final String[] servitorer = {"Mia", "Per"};
11        final int KAPASITET = 4;
12
13        //Skriv ut header
14        skrivUtHeader(kokker, servitorer, KAPASITET);
15
16        HamburgerBrett brett = new HamburgerBrett(KAPASITET);
17
18        //Starte tråder
19        for (String navn: kokker) {
20            new Kokk(brett, navn).start();
21        }
22        for (String navn: servitorer) {
23            new Servitor(brett, navn).start();
24        }
25    }
26 }
```

Run | Debug

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

I denne simuleringen har vi
3 kokker [Anne, Erik, Knut]
2 servitorer [Mia, Per]
Kapasiteten til brettet er 4 hamburgere
Vi starter...

Erik (kokk) legger til burger 1. Brett: [1]
Mia(servitor) tar burger 1. Brett []
Knut (kokk) legger til burger 2. Brett: [2]
Erik (kokk) legger til burger 3. Brett: [2, 3]
Per(servitor) tar burger 2. Brett [3]
Anne (kokk) legger til burger 4. Brett: [3, 4]
Erik (kokk) legger til burger 5. Brett: [3, 4, 5]
Mia(servitor) tar burger 3. Brett [4, 5]
Knut (kokk) legger til burger 6. Brett: [4, 5, 6]
Erik (kokk) legger til burger 7. Brett: [4, 5, 6, 7]
Mia(servitor) tar burger 4. Brett [5, 6, 7]
Erik (kokk) legger til burger 8. Brett: [5, 6, 7, 8]
Knut(kokk) vil legge på flere burgere men brettet er full, venter!
Per(servitor) tar burger 5. Brett [6, 7, 8]
Knut (kokk) legger til burger 9. Brett: [6, 7, 8, 9]
Anne(kokk) vil legge på flere burgere men brettet er full, venter!

Oppgave 3

The screenshot shows an IDE with a project named 'OBLIG1'. The file explorer on the left shows the project structure, including 'Oppgave_3' which contains 'Hamburgersjappe.java', 'Kokk.java', and 'Servitor.java'. The main editor displays the code for 'Hamburgersjappe.java'.

```
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.TimeUnit;

public class Hamburgersjappe {

    public static void main(String[] args ){

        final String[] kokker = {"Anne", "Erik", "Knut"};
        final String[] servitorer = {"Mia", "Per"};
        final int KAPASITET = 4;

        //Skriv ut header
        skrivUtHeader(kokker, servitorer, KAPASITET);

        BlockingQueue<Integer> burgerko = new LinkedBlockingQueue<Integer>();

        //Starte tråder
        for(String navn: kokker){
            new Kokk(burgerko, navn).start();
        }
        for(String navn: servitorer){
            new Servitor(burgerko, navn).start();
        }
    }

    private static void skrivUtHeader(String[] kokker, String[] servitorer, int KAPASITET) {
        System.out.println("I denne simuleringen har vi");
        System.out.println("3 kokker [Anne, Erik, Knut]");
        System.out.println("2 servitører [Mia, Per]");
        System.out.println("Kapasiteten til brettet er 4 hamburgere");
        System.out.println("Vi starter...");
    }
}
```

The terminal output shows the execution of the program:

```
lig1'; & 'C:\Program Files\Java\jdk-12.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,
spend=y,address=localhost:59336' '-cp' 'C:\Users\vlads\AppData\Roaming\Code\User\workspaceSto
10b1744d08b68fbc486668033c\redhat.java\jdt_ws\Oblig1_409d3d1c\bin' 'Oppgave_3.Hamburgersjappe
I denne simuleringen har vi
3 kokker [Anne, Erik, Knut]
2 servitører [Mia, Per]
Kapasiteten til brettet er 4 hamburgere
Vi starter...
Anne(kokk) legger på hamburger [1]. Brett: [1]
Per(servitør) tar av hamburger [1]. Brett: []
Erik(kokk) legger på hamburger [2]. Brett: [2]
Mia(servitør) tar av hamburger [2]. Brett: []
Anne(kokk) legger på hamburger [3]. Brett: [3]
Knut(kokk) legger på hamburger [4]. Brett: [3, 4]
Per(servitør) tar av hamburger [3]. Brett: [4]
Erik(kokk) legger på hamburger [5]. Brett: [4, 5]
Anne(kokk) legger på hamburger [6]. Brett: [4, 5, 6]
Mia(servitør) tar av hamburger [4]. Brett: [5, 6]
Per(servitør) tar av hamburger [5]. Brett: [6]
Anne(kokk) legger på hamburger [7]. Brett: [6, 7]
```