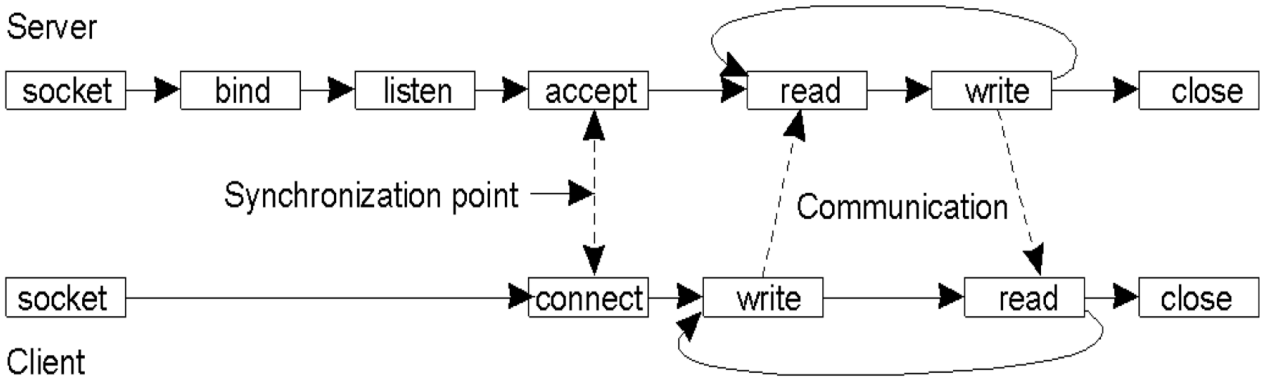
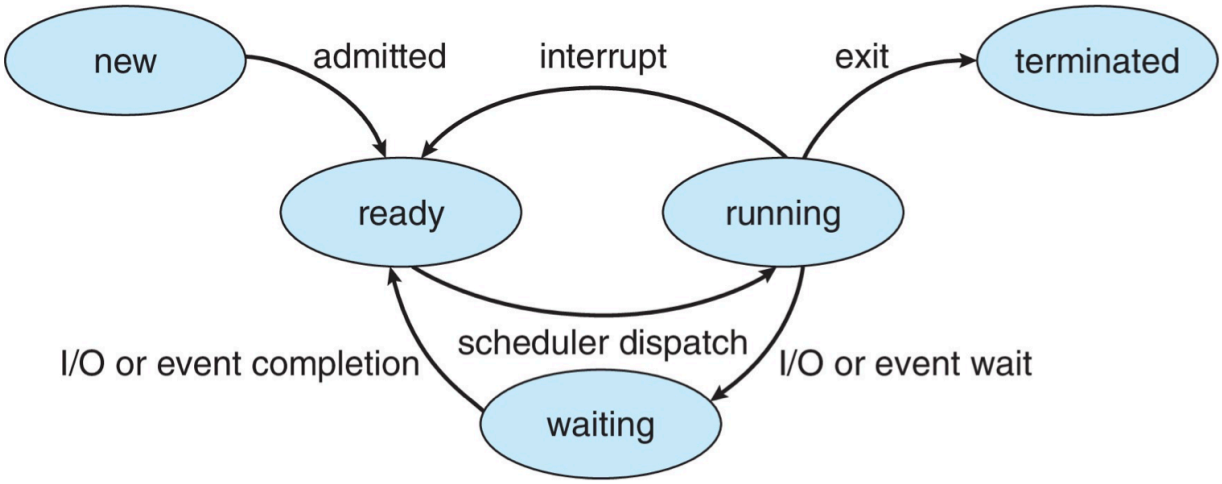
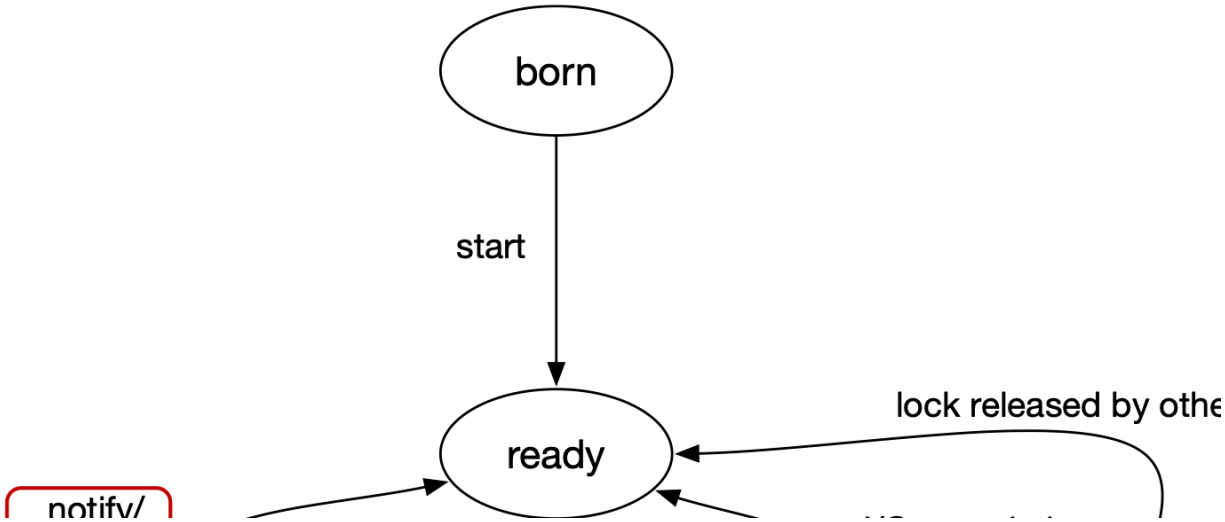
Immagini

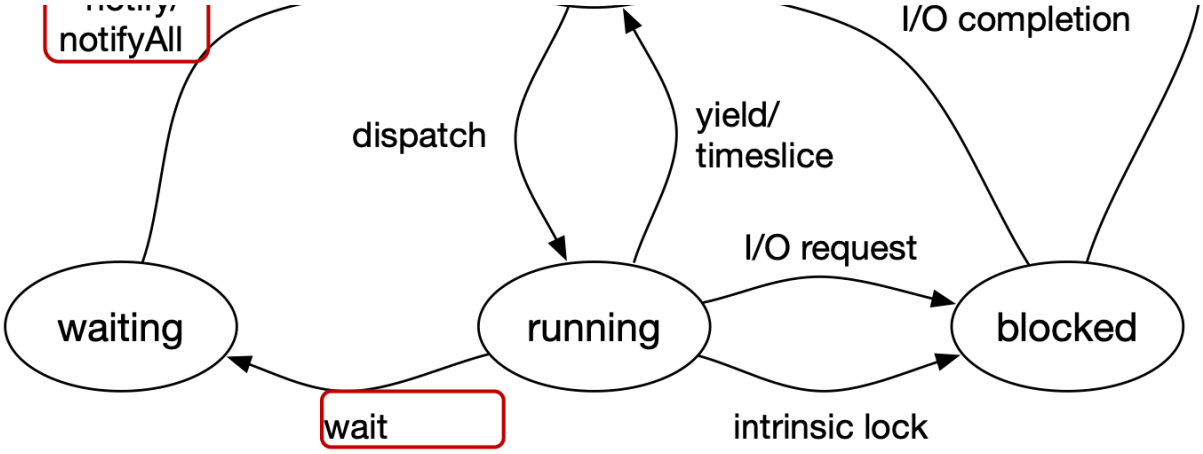
Friday, 7 July 2023 10:18

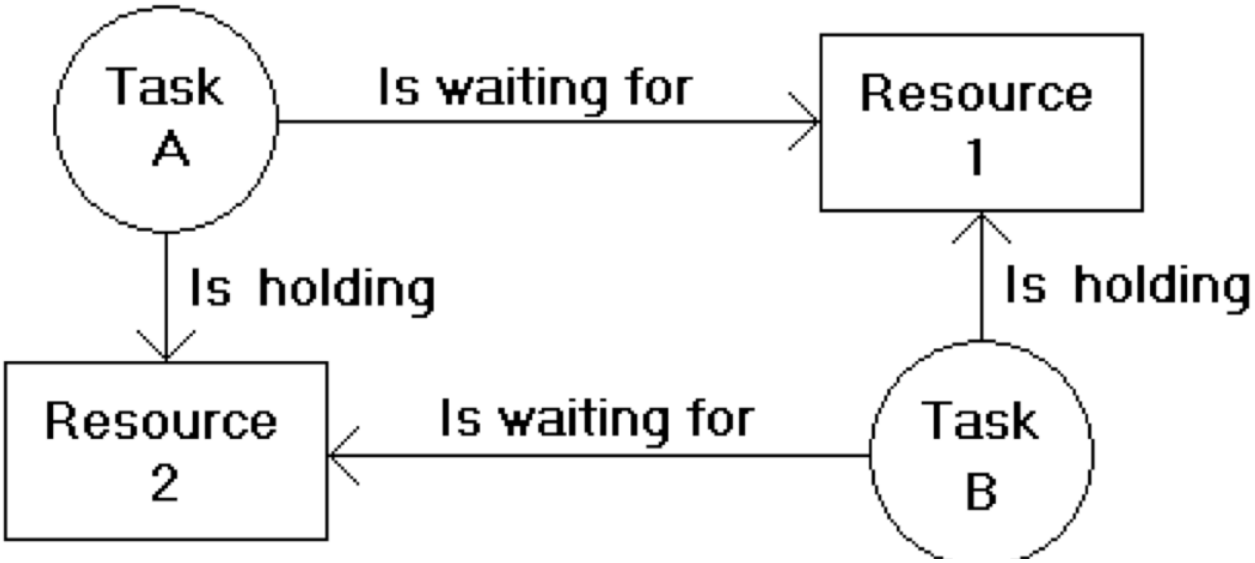
All'esame vi capiteranno minimo 3/4 domande su immagini, quindi eccovi una lista

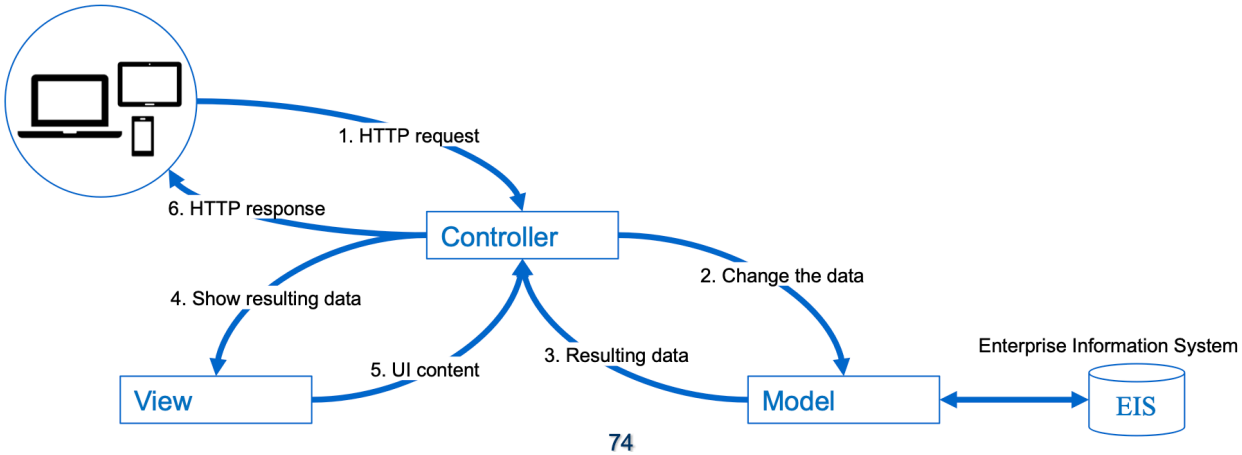




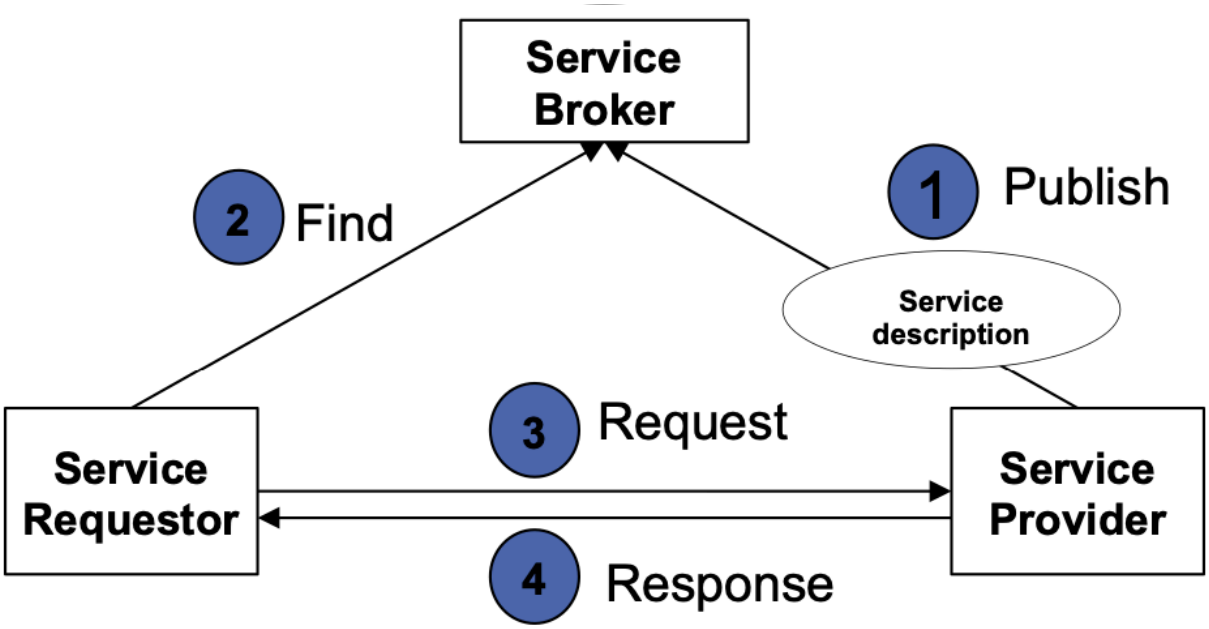



notifyAll
I/O completion
dispatch
yield/
timeslice
I/O request
waiting
running
blocked
wait
intrinsic lock

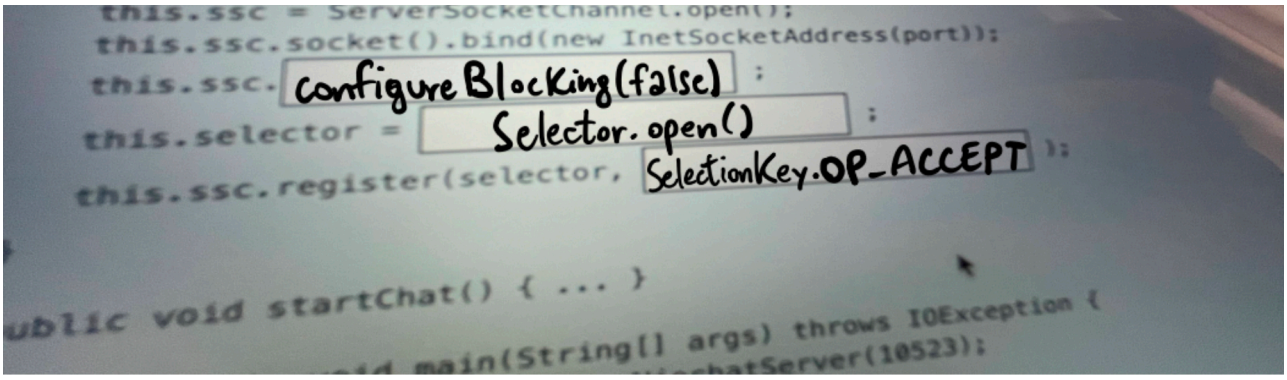

Task
A
Is waiting for
Resource
1
Is holding
Is holding
Resource
2
Is waiting for
Task
B

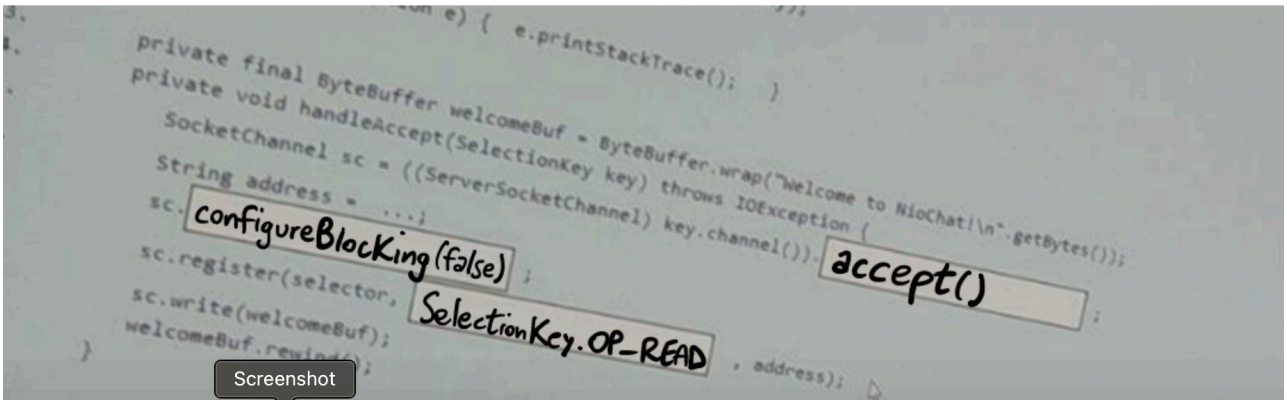

1. HTTP request
6. HTTP response
Controller
2. Change the data
4. Show resulting data
5. UI content
3. Resulting data
Enterprise Information System
View
Model
EIS

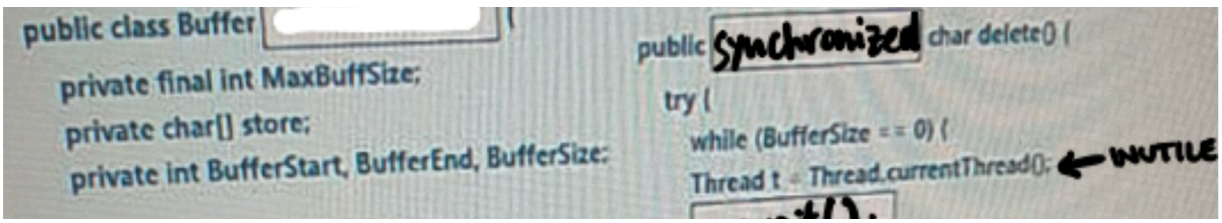

Service
Registry

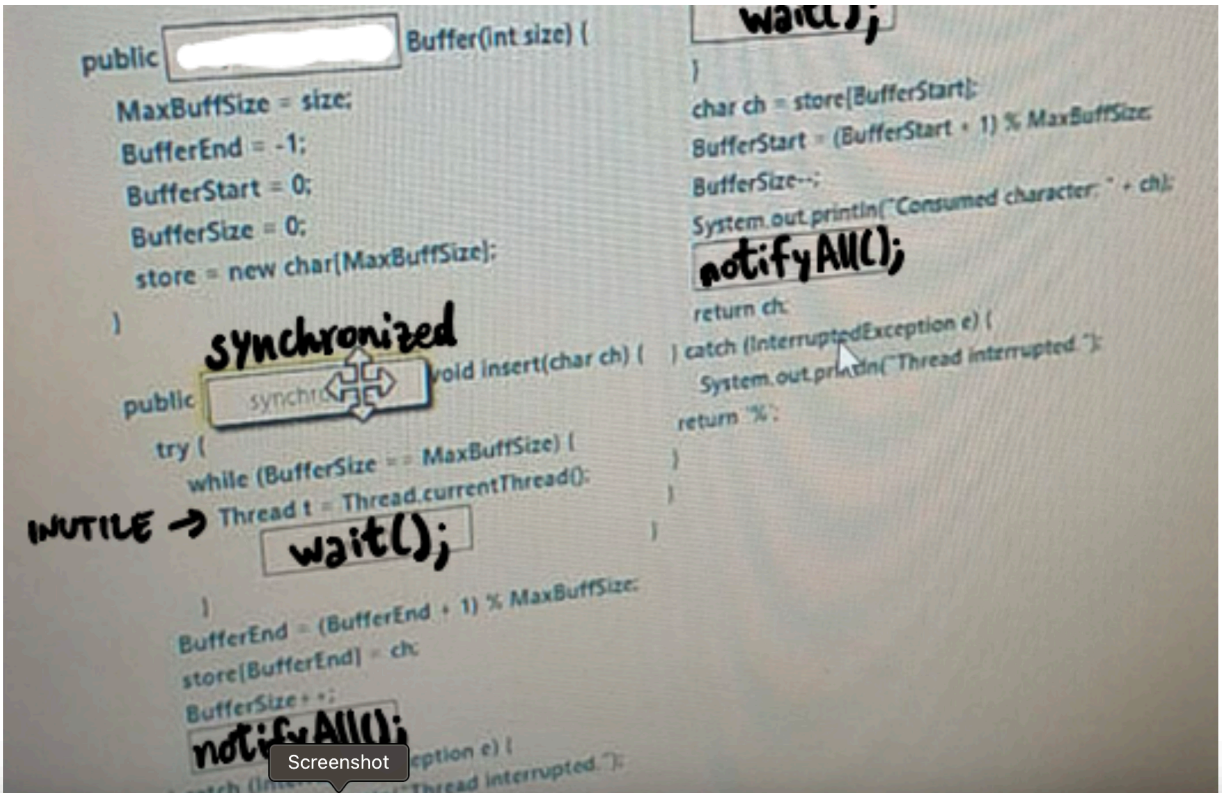


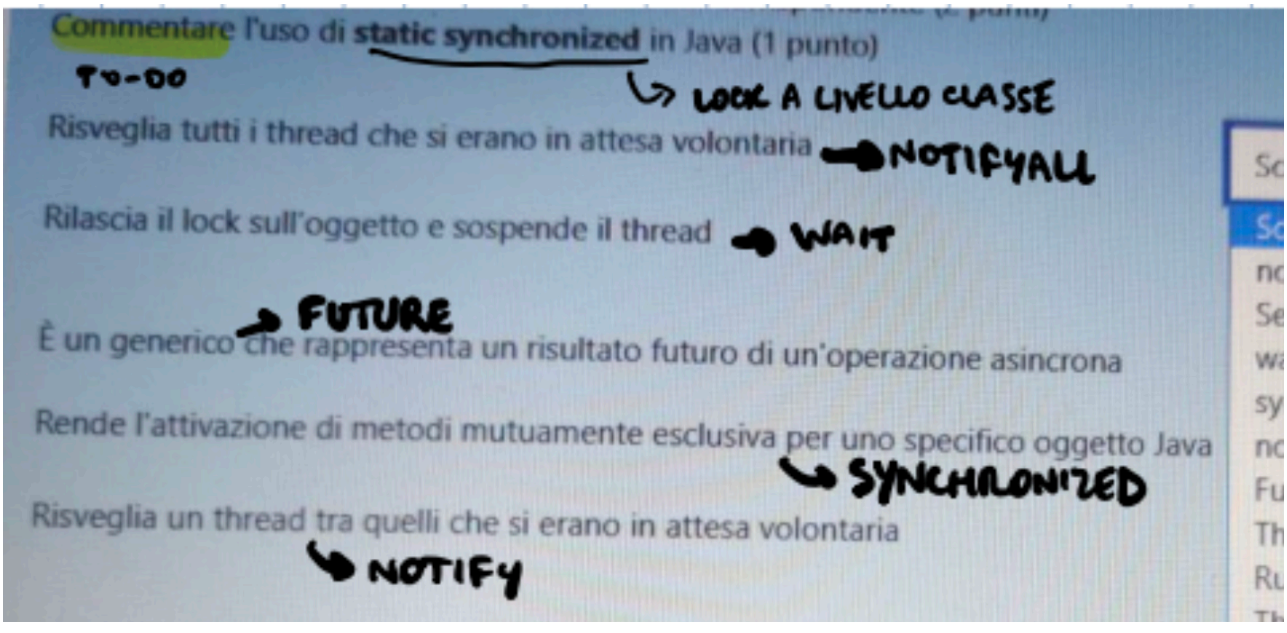
Nota: all'esame request/response sono unificati in "message", find/publish/broker hanno un nome strano italianizzaot

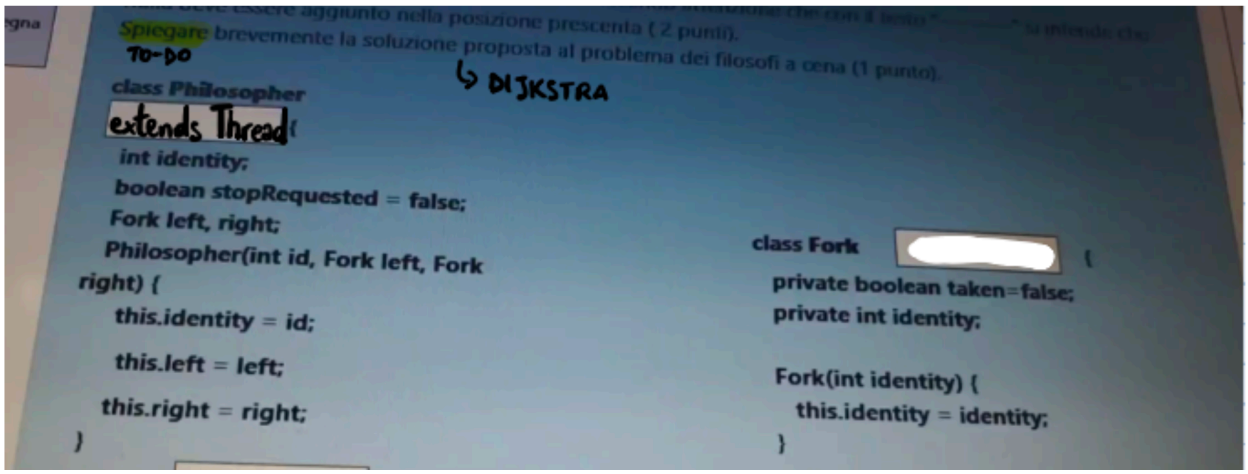




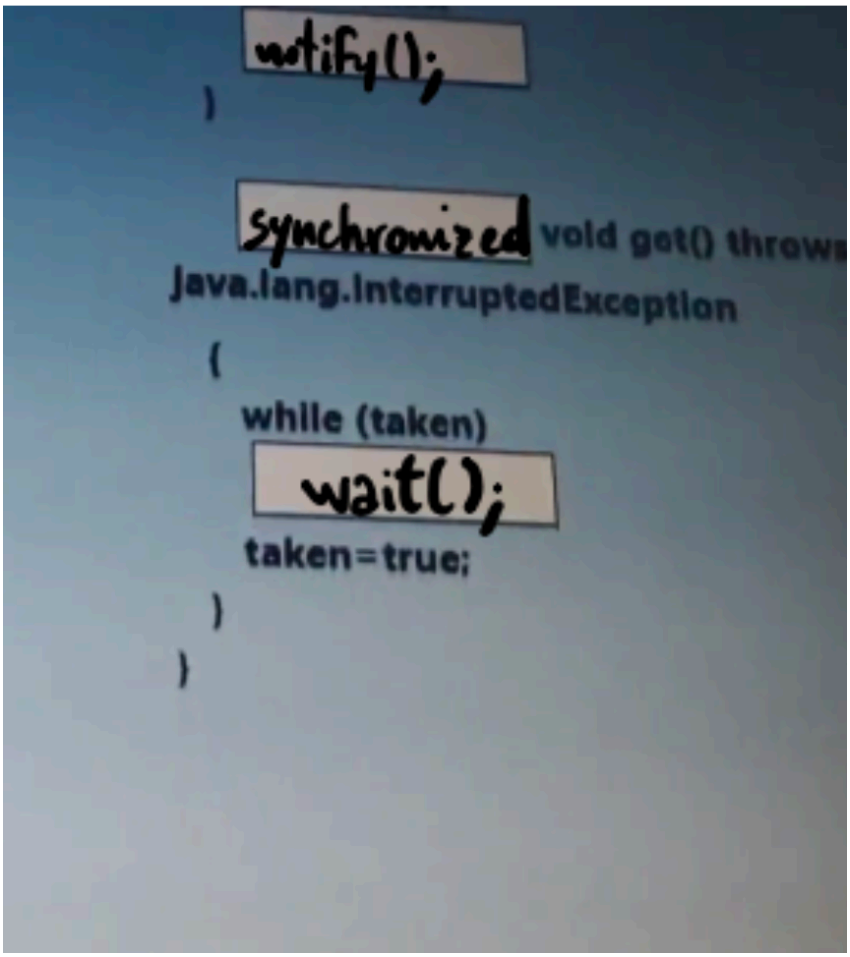


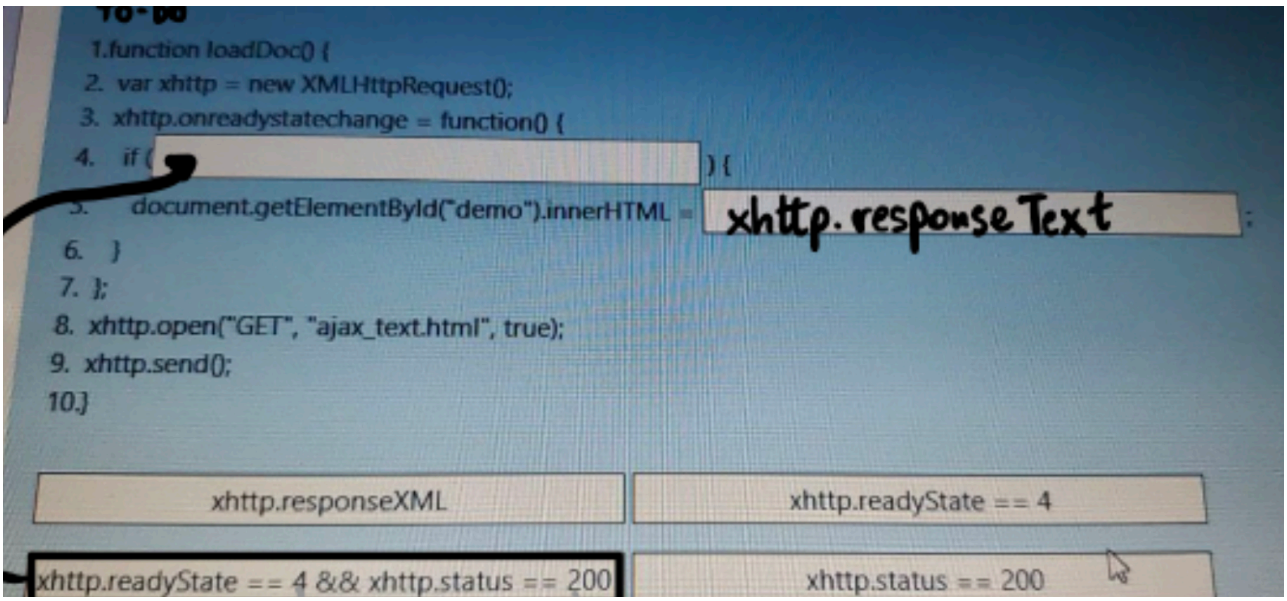

public Buffer(int size) {
MaxBuffSize = size;
BufferEnd = -1;
BufferStart = 0;
BufferSize = 0;
store = new char[MaxBuffSize];
}
synchronized
public synchronized void insert(char ch) {
try {
while (BufferSize == MaxBuffSize) {
INUTILE → Thread t = Thread.currentThread();
wait();
}
BufferEnd = (BufferEnd + 1) % MaxBuffSize;
store[BufferEnd] = ch;
BufferSize++;
notifyAll();
Screenshot
char ch = store[BufferStart];
BufferStart = (BufferStart + 1) % MaxBuffSize;
BufferSize--;
System.out.println("Consumed character: " + ch);
notifyAll();
return ch;
} catch (InterruptedException e) {
System.out.println("Thread interrupted.");
return '%';
}
}

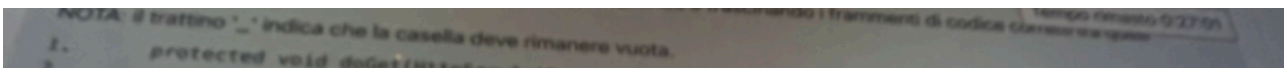

Commentare l'uso di static synchronized in Java (1 punto)
TO-DO
↳ LOCK A LIVELLO CLASSE
Risveglia tutti i thread che si erano in attesa volontaria → NOTIFYALL
Rilascia il lock sull'oggetto e sospende il thread → WAIT
→ FUTURE
È un generico che rappresenta un risultato futuro di un'operazione asincrona
Rende l'attivazione di metodi mutuamente esclusiva per uno specifico oggetto Java
↳ SYNCHRONIZED
Risveglia un thread tra quelli che si erano in attesa volontaria
↳ NOTIFY

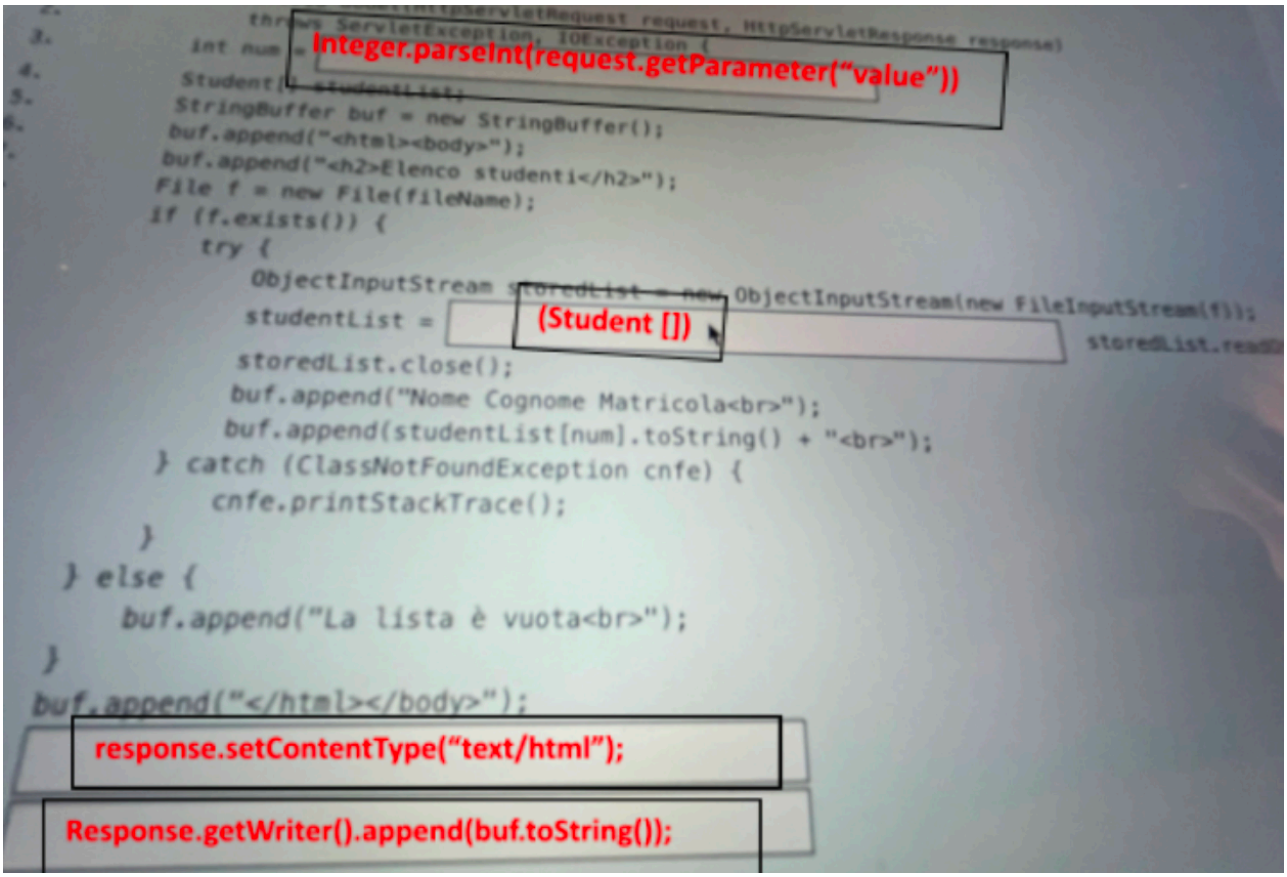

Spiegare brevemente la soluzione proposta al problema dei filosofi a cena (1 punto).
TO-DO
↳ DIJKSTRA
class Philosopher
extends Thread {
int identity;
boolean stopRequested = false;
Fork left, right;
Philosopher(int id, Fork left, Fork
right) {
this.identity = id;
this.left = left;
this.right = right;
}
class Fork {
private boolean taken=false;
private int identity;
Fork(int identity) {
this.identity = identity;
}


public
void run() {
while (!stopRequested) {
try {
//thinking
}
Synchronized void put() {
taken=false;
notify();
SE NON C'È ALLORA notifyAll


notify();
}
synchronized void get() throws
Java.lang.InterruptedException
{
while (taken)
wait();
taken=true;
}
}


1.function loadDoc() {
2. var xhttp = new XMLHttpRequest();
3. xhttp.onreadystatechange = function() {
4. if (
) {
document.getElementById("demo").innerHTML =
xhttp.responseText
;
6. }
7. };
8. xhttp.open("GET", "ajax_text.html", true);
9. xhttp.send();
10.}
xhttp.responseXML
xhttp.readyState == 4
xhttp.readyState == 4 && xhttp.status == 200
xhttp.status == 200


NOTA: il trattino '_' indica che la casella deve rimanere vuota.
protected void doGet(




```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
  String cognome = ____1____ ;
  int matricola = ____2____ ;
  Student s = new Student(cognome, matricola);
  File f = new File(fileName);
  if (f.exists()) {
    try {
      ObjectInputStream storedList = new ObjectInputStream(new FileInputStream(f));
      Student[] studentList = ____3____ storedList.readObject();
      storedList.close(); // close stream
    } catch (ClassNotFoundException cnfe) {
      cnfe.printStackTrace();
    }
    // create a new ArrayList and add the new element
    List<Student> arrlist = new ArrayList<Student>(Arrays.asList(studentList));
    arrlist.add(s);
    // Convert the Arraylist to array
    studentList = arrlist.toArray(studentList);
  } else {
    studentList = new Student[1];
    studentList[0] = s;
  }
  ObjectOutputStream output = new ObjectOutputStream(new FileOutputStream(f));
  ____4____ ;
  output.flush();
  output.close();
  doGet(request, response);
}
```

Double.parseDouble(request.getParameter("value"))	Integer.parseInt(request.getParameter("value")) 2
request.getParameter("cognome") 1	(Student)
output.println(studentList)	response.setContentType("text/html") 4
request.getParameter("value")	(Student[]) 3


Service Description
Discovery Agencies
Find
Publish
Client
Service Requestor
Interact
Service Provider
Service
Service Description