Exercise Sheet 11

1. Develop a distributed chat-application using Java and **RMI**. The server must be able to register clients and send messages to them. The client should be able to register and de-register from the server, send messages, and receive messages from the server. A message consists of a client ID, a time stamp, and a text message. You may use the socket based implementation on the course website as a framework to start with.

2. Example from the exam!

a.) Are the Java locks reentrant? What does a reentrant lock mean? Explain your answer in the context of the following Java class:

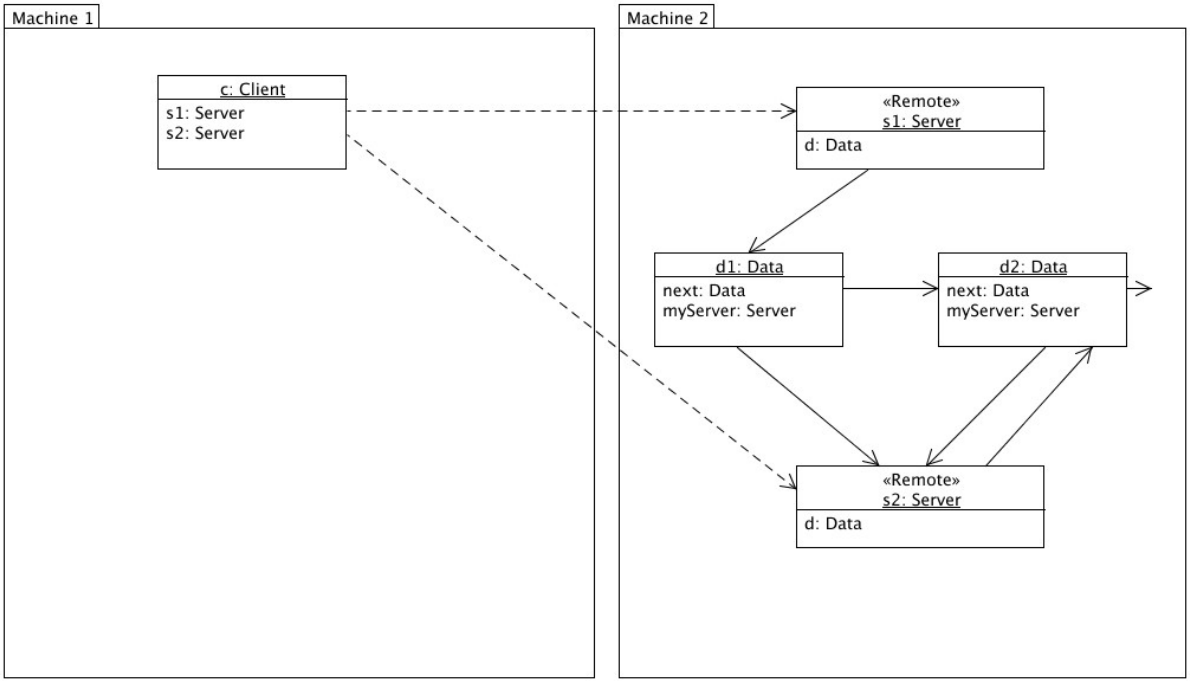
```
public class Test {
    private int f;
    public synchronized void m () {
                    if (f < 3)
                    n ();
    }

    public synchronized void n () {
                    f++;
                    m();
    }
}
```

b.) Let us assume that Java locks are not reentrant. How can the above program be rewritten without reentrant locks? Will your solution prevent deadlock?

3. Example from the exam!

The following diagram represents two machines communicating over RMI. The client is running on machine 1 and the server on machine 2. The client has only one object containing two references to `Server` objects. The server has two classes, one of type `Server` and one of type `Data`. Assume that the `Server` class has the following method: `Data getData() {return d;}`.

a) What steps happen during compilation of an RMI enabled program?
b) The client c calls the method `getData` on the server object s1. Draw the object structure on machine 1 after this call.
c) The client c calls the method `getData` on the server object s2. Draw the object structure on machine 1 after this call
d) All arguments of a remote method call are serialized together in Java. What could happen if they were serialized individually? Write a program that shows this and give a graphical representation using the diagram. The classes `Client`, `Server`, and `Data` can be used for this example.
e) Can Java RMI be referred to as location transparent? Give two examples that justify your answer.


Machine 1
c: Client
s1: Server
s2: Server
Machine 2
«Remote»
s1: Server
d: Data
d1: Data
next: Data
myServer: Server
d2: Data
next: Data
myServer: Server
«Remote»
s2: Server
d: Data