

How to bind a Message object instead of a string:

1. create a class which implements java.io.Serializable
2. implement toString method

```
import java.io.Serializable;

public class Message implements Serializable{
    String user;
    String subject;
    String content;
    public Message(String user, String subject, String content) {
        this.user = user;
        this.subject = subject;
        this.content = content;
    }
    public String toString() {
        return this.user+": <"+this.subject+"> "+this.content;
    }
}
```

```
import javax.naming.* ;
import java.util.Hashtable ;

public class JNDIExample {

    Run | Debug
    public static void main(String[] args) {
        Hashtable<String, String> env = new Hashtable<String, String>() ;
        env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory") ;
        env.put(Context.PROVIDER_URL, "ldap://localhost:10389/dc=example,dc=com") ;
        env.put(Context.SECURITY_AUTHENTICATION, "simple") ;
        env.put(Context.SECURITY_PRINCIPAL, "uid=admin, ou=system") ;
        env.put(Context.SECURITY_CREDENTIALS, "secret") ;
        try {
            Context ctx = new InitialContext(env) ;
            Message message = new Message("Leo", "COMP3358 Tutorial 4", "To test the Message object.");
            ctx.rebind( "cn=test", message) ;

            Message msg = (Message) ctx.lookup( "cn=test") ;
            System.out.println(msg) ;
        } catch (NamingException e) {
            e.printStackTrace() ;
        }
    }
}
```

result:

```
(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4$ java JNDIExample
[phj] T4 <sep> Testing!
(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4$
```

```
anacron
apport
(base) phj@phjs-comp:~$ sudo /etc/init.d/a
apacheds-2.0.0.AM26-default avahi-daemon
acpid
alsa-utils
apparmor
anacron
apport
(base) phj@phjs-comp:~$ sudo /etc/init.d/apacheds-2.0.0.AM26-default start
[sudo] password for phj:
Starting ApacheDS - default...
Removed stale pid file: /var/lib/apacheds-2.0.0.AM26/default/run/apacheds-default.pid
(base) phj@phjs-comp:~$
```

The screenshot shows the LDAP Browser interface. The left pane displays the directory tree with 'cn=test,dc=example,dc=com' selected. The main pane shows the search results for 'cn=test,dc=example,dc=com'. The search results table lists the following attributes and values:

Attribute	Description	Value
objectClass	javaContainer (structural)	
objectClass	javaObject (abstract)	
objectClass	javaSerializedObject (auxiliary)	
objectClass	top (abstract)	
cn		test
javaClassName		Message

The search logs at the bottom show the following details:

```
#!DATE 2021-04-07T13:57:34.783
# LDAP URL : ldap://127.0.0.1:10389/dc=example,dc=com?hasSubordinates,objectClass?one?(objectClass=*)
# command line : ldapsearch -H ldap://127.0.0.1:10389 -x -D "uid=admin,ou=system" -W -b "dc=example,dc=com" "(cn=test)"
# baseObject : dc=example,dc=com
# scope : singleLevel (1)
# derefAliases : derefAlways (3)
# sizeLimit : 1000
# timeLimit : 0
# typesOnly : False
# filter : (objectClass=*)
# attributes : hasSubordinates objectClass
```

Exercise 2:

what need to do to modify the client?

1. create the Hash table env
2. get the TicketMachine class use the Content lookup method

```
Run | Debug
public static void main(String [] args) throws RemoteException, MalformedURLException, NamingException {
    Hashtable<String, String> env = new Hashtable<String, String>() ;
    env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
    env.put(Context.PROVIDER_URL, "ldap://localhost:10389/dc=example,dc=com");
    env.put(Context.SECURITY_AUTHENTICATION, "simple");
    env.put(Context.SECURITY_PRINCIPAL, "uid=admin, ou=system");
    env.put(Context.SECURITY_CREDENTIALS, "secret");
    String rmiurl = "rmi://localhost/TicketMachine";
    Context ctx = new InitialContext(env);
    TicketMachine app = new TicketServer();

    Reference ref = new Reference("TicketMachine", new StringRefAddr("URL", rmiurl));

    ctx.rebind("cn=TicketMachine", ref);
    ctx.rebind(rmiurl, app);

    System.out.println("Service registered");
}

private int counter;
public TicketServer() throws RemoteException {
    counter = 1000;
}
public int getTicket() throws RemoteException {
    return counter++;
}
```

```
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.util.Hashtable;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

public class Customer {
    Run | Debug
    public static void main(String[] args) throws MalformedURLException, RemoteException, NotBoundException, NamingException {

        Hashtable<String, String> env = new Hashtable<String, String>() ;

        env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
        env.put(Context.PROVIDER_URL, "ldap://localhost:10389/dc=example,dc=com");
        env.put(Context.SECURITY_AUTHENTICATION, "simple");
        env.put(Context.SECURITY_PRINCIPAL, "uid=admin, ou=system");
        env.put(Context.SECURITY_CREDENTIALS, "secret");

        Context ctx = new InitialContext(env);
        TicketMachine app = (TicketMachine)ctx.lookup("cn=TicketMachine");
        int ticket = app.getTicket();

        System.out.println("Got ticket number: " + ticket);
    }
}
```

result:

to get the result you need to set up the environment just like tutorial 3

```
^C(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4/rmiexample$ rmiregistry
```

```
(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4/rmiexample$ java TicketServer -Djava.security.policy=security.policy
Service registered
```

```
(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4/rmiexample$ ls
Customer.class Customer.java security.policy TicketMachine.class TicketMachine.java TicketServer.class TicketServer.java
(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4/rmiexample$ java Customer
Got ticket number: 1000
(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4/rmiexample$ java Customer
Got ticket number: 1001
(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4/rmiexample$ java Customer
Got ticket number: 1002
(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4/rmiexample$ java Customer
Got ticket number: 1003
(base) phj@phjs-comp:~/Downloads/3358/tutorial/T4/rmiexample$
```