

## Lab 6

### Creating a registration module

**Objective:** To study the capabilities of the javax.security.auth library

**Objective:** Create a registration module using the javax.security.auth library

**Work order:**

1. Explore the methods and capabilities of the javax.security.auth library
2. Create your own registration module
3. Check the functionality of the module

**Work progress:**

1. Creation of the registration module implies the usage of several classes and interfaces of the javax.security.auth library.

The first of these interfaces is the LoginModule itself. This interface allows the implementation of various types of authentication technologies.

This interface contains the initialize() method, which creates the interface itself, the checkLogin() method, which checks the login and password pairs, and the login() method, which checks the correctness of the pair's input and allows or does not allow further. The second interface is CallbackHandler, which connects the LoginModule and the application. It serves to display certain authentication information or to display a message.

The handler DialogCallbackHandler opens a simple GUI dialog box to prompt for a username and password, and the handler javax.security.auth.callback.TextCallbackHandler retrieves these credentials from the console.

2. To do this, you need to create your own classes that use these interfaces.

a) So, for example, create a SimpleLoginModule class that uses the interface

```
LoginModule
public class SimpleLoginModule implements LoginModule
{
    private Subject subject;
    private CallbackHandler callbackHandler;
    private Map<String, ?> options;
    public void initialize(Subject subject, CallbackHandler
        callbackHandler,
        Map<String, ?> sharedState, Map<String, ?> options)
    {
        this.subject = subject;
        this.callbackHandler = callbackHandler;
        this.options = options;
    }
}
```

```

    }
}

```

b) Create a *SimpleCallbackHandler* class using the interface *CallbackHandler*

```

public class SimpleCallbackHandler implements CallbackHandler
{
    private String username;
    private char[] password;

    public SimpleCallbackHandler(String username, char[] password)
    {
        this.username = username;
        this.password = password;
    }
}

```

3. Create a form that implements the registration module

```

public class JAASFrame extends JFrame
{
    private JTextField username;
    private JPasswordField password;
    private JTextField propertyName;
    private JTextField propertyValue;

    public JAASFrame()
    {
        username = new JTextField(20);
        password = new JPasswordField(20);
        propertyName = new JTextField("user.home");
        propertyValue = new JTextField(20);
        propertyValue.setEditable(false);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(0, 2));
        panel.add(new JLabel("username:"));
        panel.add(username);
        panel.add(new JLabel("password:"));
        panel.add(password);
        panel.add(propertyName);
        panel.add(propertyValue);
        add(panel, BorderLayout.CENTER);

        JButton getValueButton = new JButton("Get Value");

        getValueButton.addActionListener(EventHandler.create(ActionListen
er.class, this, "getValue"));
        JPanel buttonPanel = new JPanel();
        buttonPanel.add(getValueButton);
    }
}

```

```

        add(buttonPanel, BorderLayout.SOUTH);
        pack();
    }

    public void getValue()
    {
        try
        {
            LoginContext context = new LoginContext("Login", new
            SimpleCallbackHandler(
                username.getText(), password.getPassword()));
            System.out.println("Trying to log in with " +
            username.getText() + " and " + new
            String(password.getPassword()));
            context.login();
            Subject subject = context.getSubject();
            propertyValue.setText("
                + Subject.doAsPrivileged(subject, new
            SysPropAction(propertyName.getText(), null));
            context.logout();
        }
        catch (LoginException ex)
        {
            ex.printStackTrace();
            Throwable ex2 = ex.getCause();
            if (ex2 != null) ex2.printStackTrace();
        }
    }
}

```



4. Check the functionality of the registration module