

**** 정의 ****

서버프로그램 구현이란 애플리케이션 설계를 기반으로 개발에 필요한 **환경을 구성하고**, 프로그래밍 언어와 도구를 활용하여 **공통모듈, 업무프로그램과 배치 프로그램**을 구현하는 능력이다.

**** 수준 5 ****

*** 학습시간 : 70시간 (2019.07.01 월 7교시 ~ 2019.07.12 금 4교시) ***

작성자 : 김 문 선

[2001020211_16v4.1 개발환경 구축하기]

*** 수행준거

- 1.1 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어의 필요 사항을 검토하고 이에 따라, 개발환경에 필요한 준비를 수행할 수 있다.
- 1.2 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어를 **설치하고 설정하여 개발환경을 구축할 수 있다.**
- 1.3 사전에 수립된 형상관리 방침에 따라, 운영정책에 부합하는 **형상관리 환경을** 구축할 수 있다.

[2001020211_16v4.2 공통 모듈 구현하기]

*** 수행준거

- 2.1 공통 모듈의 상세 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 업무 프로세스 및 서비스의 구현에 필요한 공통 모듈을 작성할 수 있다.
- 2.2 소프트웨어 측정지표 중 **모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높인 공통모듈을** 구현할 수 있다.
- 2.3 개발된 공통 모듈의 내부 기능과 제공하는 인터페이스에 대해 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.

[2001020211_16v4.3 서버 프로그램 구현하기]

*** 수행준거

- 3.1 업무 프로세스 맵과 세부 업무 프로세스를 확인할 수 있다.
- 3.2 세부 업무프로세스를 기반으로 프로그래밍 언어와 도구를 활용하여 서비스의 구현에 필요한 업무 프로그램을 구현할 수 있다.
- 3.3 개발하고자 하는 목표 시스템의 잠재적 보안 취약성이 제거될 수 있도록 서버 프로그램을 구현할 수 있다.
- 3.4 개발된 업무 프로그램의 내부 기능과 제공하는 인터페이스에 대해 테스트를 수행할 수 있다.

[2001020211_16v4.4 배치 프로그램 구현하기]

*** 수행준거

- 4.1 애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 구현 할 수 있다.
- 4.2 목표 시스템을 구성하는 하위 시스템간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 구현 할 수 있다.
- 4.3 개발된 배치 프로그램을 테스트를 수행할 수 있다.

[2001020211_16v4.1 개발환경 구축하기]

=> 서버프로그램을 구현하기 위해 개발환경을 구축합니다.

1. JDK 설치하기

이름	수정된 날짜	유형	크기
apache-tomcat-8.5.37	2019-07-08 오전...	파일 폴더	
eclipse	2019-07-08 오전...	파일 폴더	
apache-tomcat-8.5.37-windows-x64.zip	2019-01-28 오후...	ALZip ZIP File	11,309KB
eclipse-jee-2018-12-R-win32-x86_64.zip	2018-12-27 오전...	ALZip ZIP File	342,017KB
jdk-8u171-windows-x64.exe	2018-05-24 오전...	응용 프로그램	212,249KB
jdk-8u191-windows-x64.exe	2018-12-27 오후...	응용 프로그램	212,193KB

** 환경변수 설정하기

=> 바탕화면->내 컴퓨터->속성->고급->환경변수->시스템변수

① 새로 만들기 -> 변수이름 : JAVA_HOME

변수값: C:\Program Files\Java\jdk1.8.0_211

② Path

변수이름 : Path





변수값 : %JAVA_HOME%\bin

[2001020211_16v4 서버프로그램 구현]

[2001020211_16v4.1 개발환경 구축하기]

=> 서버프로그램을 구현하기 위해 개발환경을 구축합니다.

2. eclipse 설치하기

이름	수정된 날짜	유형	크기
 apache-tomcat-8.5.37-windows-x64.zip	2019-01-28 오후...	ALZip ZIP File	11,309KB
 eclipse-jee-2018-12-R-win32-x86_64.zip	2018-12-27 오전...	ALZip ZIP File	342,017KB
 jdk-8u171-windows-x64.exe	2018-05-24 오전...	응용 프로그램	212,249KB
 jdk-8u191-windows-x64.exe	2018-12-27 오후...	응용 프로그램	212,193KB

** 압축을 풀고 "eclipse.exe" 파일 실행

이름	수정된 날짜	유형	크기
 configuration	2019-07-12 오전...	파일 폴더	
 dropins	2018-12-14 오전...	파일 폴더	
 features	2019-07-08 오후...	파일 폴더	
 p2	2019-07-12 오전...	파일 폴더	
 plugins	2019-07-08 오후...	파일 폴더	
 readme	2019-07-08 오후...	파일 폴더	
 .eclipseproduct	2018-09-06 오후...	ECLIPSEPRODUCT...	1KB
 artifacts.xml	2018-12-14 오전...	XML 문서	276KB
 eclipse.exe	2018-12-14 오전...	응용 프로그램	415KB
 eclipse.ini	2018-12-14 오전...	구성 설정	1KB
 eclipsesec.exe	2018-12-14 오전...	응용 프로그램	127KB
 컴개발용.exe	2018-12-14 오전...	응용 프로그램	415KB

[2001020211_16v4 서버프로그램 구현]

[2001020211_16v4.1 개발환경 구축하기]

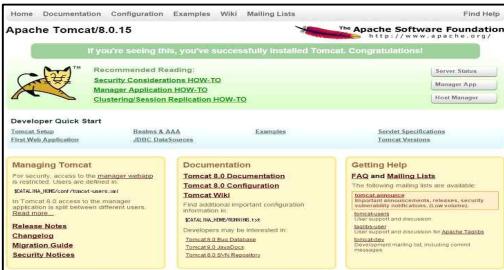
=> 서버프로그램을 구현하기 위해 개발환경을 구축합니다.

3. apache-tomcat 설치하기

이름	수정된 날짜	유형	크기
 apache-tomcat-8.5.37-windows-x64.zip	2019-01-28 오후...	ALZip ZIP File	11,309KB
 eclipse-jee-2018-12-R-win32-x86_64.zip	2018-12-27 오전...	ALZip ZIP File	342,017KB
 jdk-8u171-windows-x64.exe	2018-05-24 오전...	응용 프로그램	212,249KB
 jdk-8u191-windows-x64.exe	2018-12-27 오후...	응용 프로그램	212,193KB

** 압축을 풀고 정상적으로 설치되었는지 확인

**>> localhost:8080/



Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.0.15

The Apache Software Foundation
http://www.apache.org/

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
[Security Considerations HOW-TO](#)
[Manager Application HOW-TO](#)
[Clustering/Session Replication HOW-TO](#)

[Server Status](#)
[Manager App](#)
[Host Manager](#)

Developer Quick Start
[Tomcat Setup](#)
[First Web Application](#)
[Realms & AAA](#)
[JDBC DataSources](#)
[Examples](#)
[Secret Specifications](#)
[Tomcat Versions](#)

Managing Tomcat
 For security, access to the [manager webapp](#) is restricted. Users are defined in:
 \$CATALINA_HOME/conf/tomcat-users.xml
 In Tomcat 8.0 access to the manager application is split between different users.
[Read more...](#)

Release Notes
[Changelog](#)
[Migration Guide](#)
[Security Notices](#)

Documentation
[Tomcat 8.0 Documentation](#)
[Tomcat 8.0 Configuration](#)
[Tomcat Wiki](#)
 Find additional important configuration information in:
 \$CATALINA_HOME/README.txt
 Developers may be interested in:
[Tomcat 8.0 Bug Database](#)
[Tomcat 8.0 JavaDoc](#)
[Tomcat 8.0 SVN Repository](#)


Getting Help
FAQ and Mailing Lists
 The following mailing lists are available:
[tomcat-announce](#)
 Important announcements, releases, security vulnerability notifications, (low volume).
[tomcat-users](#)
 User support and discussion.
[tomcat-dev](#)
 User support and discussion for Apache Tomcat.
[tomcat-dev](#)
 Development mailing list, including commit messages.

[2001020211_16v4 서버프로그램 구현]

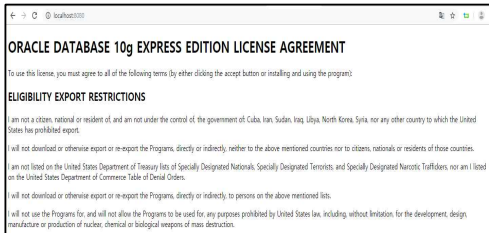
[2001020211_16v4.1 개발환경 구축하기]

=> 서버프로그램을 구현하기 위해 개발환경을 구축합니다.

4. oracle 설치하기

이름	수정된 날짜	유형	크기
 ojdbc14.jar	2006-02-02 오전...	ALZip JAR File	1,501KB
 ojdbc14-10.2.0.4.0.jar	2011-09-27 오전...	ALZip JAR File	1,520KB
 OracleXEUniv.exe	2014-11-24 오후...	응용 프로그램	211,849KB
 sqldeveloper-4.2.0.17.089.1709-no-jre.z...	2017-04-14 오후...	ALZip ZIP File	340,435KB
 staruml-5.0-with-cm.exe	2017-03-21 오전...	응용 프로그램	22,194KB

** 설치 후, 오라클 정상적으로 설치되었는지 확인



[2001020211_16v4 서버프로그램 구현]

[2001020211_16v4.2 공통 모듈 구현하기]

=> 공통 모듈의 상세 설계 산출물을 기반으로 공통 모듈을 구현합니다.

1. 공통 모듈을 구현하기 위한 I/O 오브젝트(Value Object)를 정의 파일 "memberVO"를 생성

```
package basic;

public class memberVO {
    private String id ;
    private String password ;
    private String name ;
    private String lev ;
    private int pay;
    private String birthd ;
    private double weight ;

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getLev() {
        return lev;
    }
    public void setlev(String lev) {
        this.lev = lev;
    }
    public int getPay() {
        return pay;
    }
}
```

```
public void setPay(int pay) {
    this.pay = pay;
}
public String getBirthd() {
    return birthd;
}
public void setBirthd(String birthd) {
    this.birthd = birthd;
}
public double getWeight() {
    return weight;
}
public void setWeight(double weight) {
    this.weight = weight;
}
public String toString() { }
```

**** 공통 코드 조회를 위해 필요한 변수들을 정의하여
계층 간 데이터 교환을 위해 사용한다**

[2001020211_16v4.2 공통 모듈 구현하기]

=> 공통 모듈의 상세 설계 산출물을 기반으로 공통 모듈을 구현합니다.

2. 데이터 접근 객체(DAO: Data Access Object)를 구현

```
package basic;

import java.sql.Connection;

public class memberDAO {
    private Connection cn = JDBCUtil.getConnection();
    private Statement st ;
    private PreparedStatement pst ;
    private ResultSet rs;
    private String sql;

    // memberList() : 전체 조회.
    public ArrayList<memberVO> memberList() {
        sql = "select * from member";
        ArrayList<memberVO> mList = new ArrayList<memberVO>();
        try {
            st = cn.createStatement();
            rs = st.executeQuery(sql);
            if(rs.next()) {
                do {
                    memberVO vo = new memberVO();
                    vo.setId(rs.getString(1));
                    vo.setPassword(rs.getString(2));
                    vo.setName(rs.getString(3));
                    vo.setLev(rs.getString("lev"));
                    vo.setPay(rs.getInt("pay"));
                    vo.setBirthd(rs.getString(6));
                    vo.setWeight(rs.getDouble(7));
                    mList.add(vo);
                } while(rs.next());
            } else {
                System.out.println("올려달 memberList 자료가 1건도 없습니다.");
                mList = null ;
            }
        } catch(Exception e) {
            System.out.println("memberList Exception>" + e.toString());
            mList = null ;
        }
        return mList;
    }
}
```

```
public memberVO loginCheck(memberVO vo) { // member 로그인
    sql = "select * from member where id = ? and password= ?" ; // ?
    try {
        pst = cn.prepareStatement(sql);
        pst.setString(1, vo.getId());
        pst.setString(2, vo.getPassword());
        rs = pst.executeQuery();
        if (rs.next()) {return vo;}
    } catch (Exception e) {
        System.out.println("memberLogin Exception>" + e.toString());
    }
    return null;
}
```

**** DAO는 SQL문을 구현하고 해당 ID값을 호출하여 데이터를 조회하거나 조작한다.**

[2001020211_16v4 서버프로그램 구현]

[2001020211_16v4.2 공통 모듈 구현하기]

=> 공통 모듈의 상세 설계 산출물을 기반으로 공통 모듈을 구현합니다.

3. 컨트롤러(Controller)를 구현 _ 01) ListController.java

```
package mController;

import java.io.IOException;

@WebServlet("/mlist")
public class ListController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public ListController() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // DAO의 memberList 호출
        memberDAO mdao = new memberDAO();
        ArrayList<memberVO> list = mdao.memberList();
        request.setAttribute("banana", list);
        // 출력은 작성 -> 웹브라우저 : jsp가 처리
        request.getRequestDispatcher("/member/memberList.jsp").forward(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}
```

[2001020211_16v4 서버프로그램 구현]

[2001020211_16v4.2 공통 모듈 구현하기]

=> 공통 모듈의 상세 설계 산출물을 기반으로 공통 모듈을 구현합니다.

3. 컨트롤러(Controller)를 구현 _ 02) LoginController.java

```
package mController;
import java.io.IOException;

@WebServlet("/login")
public class LoginController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public LoginController() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Form에서 전송된 값을 처리
        // -> request.get.... => vo에 담기
        memberVO vo = new memberVO();
        String id = request.getParameter("id");
        String password = request.getParameter("password");
        vo.setId(id);
        vo.setPassword(password);

        // DAO의 memberLogin 호출
        memberDAO mdao = new memberDAO();
        //vo = mdao.loginCheck(vo);
        vo = mdao.loginCheck(vo);
        if(vo!=null) {
            // 로그인 성공
            HttpSession session = request.getSession();
            session.setAttribute("loginID", vo.getId());
            request.getRequestDispatcher("/login/loginSuccess.jsp").forward(request, response);
        } else {
            // 로그인 실패
            request.getRequestDispatcher("/login/loginFail.jsp").forward(request, response);
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}
```

[2001020211_16v4 서버프로그램 구현]

[2001020211_16v4.3 서버 프로그램 구현하기]

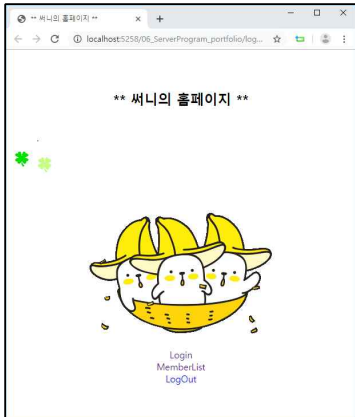
=> 세부 업무 프로세스를 기반으로 업무 프로그램을 구현합니다.

1. 먼저, index.jsp 서버 프로그램의 main 화면을 실행합니다.

```
<% page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>** 씨니의 홈페이지 **</title>
<style type="text/css">
    a {
        text-decoration : none;
    }
    body {
        text-align:center;
    }
</style>
</head>
<body>
<br>
<br>
<h2>** 씨니의 홈페이지 **</h2>
<br>
<marquee width="500px">Welcome...</marquee>

<br>
<img src = "../image/ban.gif"><br>
<img src = "../image/banana2.gif"><br>
<a href = "/06_ServerProgram_portfolio/Login/LoginForm.jsp">Login</a><br>
<a href = "/06_ServerProgram_portfolio/mlist">MemberList</a><br>
<a href = "/06_ServerProgram_portfolio/Login/LogOut.jsp">LogOut</a><br>

</body>
</html>
```



[2001020211_16v4 서버프로그램 구현]

[2001020211_16v4.3 서버 프로그램 구현하기]

=> 세부 업무 프로세스를 기반으로 업무 프로그램을 구현합니다.

2. "Login"을 클릭하여 로그인 화면을 실행합니다.

```
<% page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>** Login Form **</title>
</head>
<body>
<center>
<form action="/06_ServerProgram_portfolio/Login" method="post">
<table>
<tr>
<td>I D : </td>
<td><input type="text" name="id" value="banana"></td>
</tr>
<tr>
<td>P W : </td>
<td><input type="password" name="password"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="login"> &nbsp; <input type="reset" value="Reset">
</td>
</tr>
</table>
</form>
</center>
</body>
</html>
```

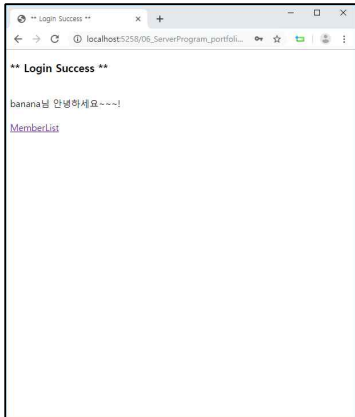
The screenshot shows a web browser window titled "Login Form". The address bar displays "localhost:5258/06_ServerProgram_portfolio/log...". The page content includes a form with two input fields: "I D : banana" and "P W :". Below these fields are two buttons labeled "login" and "Reset".

[2001020211_16v4.3 서버 프로그램 구현하기]

=> 세부 업무 프로세스를 기반으로 업무 프로그램을 구현합니다.

3. 물 바른 아이디와 패스워드 입력 시, 성공한 화면을 보여줍니다. (ID: banana / PW: 12345)

```
<% page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>** Login Success **</title>
</head>
<body>
<h3>** Login Success **</h3>
<br>
${loginID}님 안녕하세요~~~!<br><br>
<a href = "/06_ServerProgram_portfolio/mlist">MemberList</a>
</body>
</html>
```



[2001020211_16v4 서버프로그램 구현]

[2001020211_16v4.3 서버 프로그램 구현하기]

=> 세부 업무 프로세스를 기반으로 업무 프로그램을 구현합니다.

3 - 1. 로그인 성공 시, "MemberList" 클릭 -> member 테이블에 있는 내용들을 확인합니다.

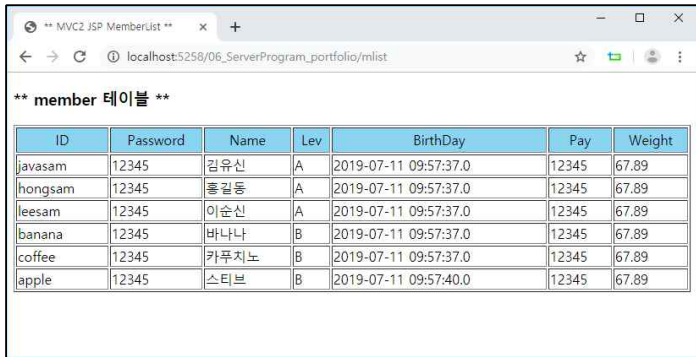
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>** MVC2 JSP MemberList **</title>
<style type = "text/css">
    tr {
        color : "blue";
    }
    body{
        text-align:"center";
    }
</style>
</head>
<body>
<h3>** member 테이블 **</h3>
<table width = 800 border="1">
<tr align = "center" height = 30 bgcolor="skyblue">
    <td>ID</td><td>Password</td><td>Name</td>
    <td>Lev</td><td>BirthDay</td><td>Pay</td>
    <td>Weight</td>
</tr>
<c:forEach var="mm" items="${banana}">
    <tr>
        <td>${mm.id}</td><td>${mm.password}</td><td>${mm.name}</td>
        <td>${mm.lev}</td><td>${mm.birthd}</td><td>${mm.pay}</td>
        <td>${mm.weight}</td>
    </tr>
</c:forEach>
</table>
</body>
</html>
```

=> 소스코드

[2001020211_16v4.3 서버 프로그램 구현하기]

=> 세부 업무 프로세스를 기반으로 업무 프로그램을 구현합니다.

3 - 1. 로그인 성공 시, "MemberList" 클릭 -> member 테이블에 있는 내용들을 확인합니다.



**** member 테이블 ****

ID	Password	Name	Lev	BirthDay	Pay	Weight
javasam	12345	김유신	A	2019-07-11 09:57:37.0	12345	67.89
hongsam	12345	홍길동	A	2019-07-11 09:57:37.0	12345	67.89
leesam	12345	이순신	A	2019-07-11 09:57:37.0	12345	67.89
banana	12345	바나나	B	2019-07-11 09:57:37.0	12345	67.89
coffee	12345	카푸치노	B	2019-07-11 09:57:37.0	12345	67.89
apple	12345	스티브	B	2019-07-11 09:57:40.0	12345	67.89

=> 실행화면

[2001020211_16v4 서버프로그램 구현]

[2001020211_16v4.3 서버 프로그램 구현하기]

=> 세부 업무 프로세스를 기반으로 업무 프로그램을 구현합니다.

4. 틀린 아이디와 패스워드 입력 시, 실패 화면을 보여줍니다. (재 로그인 화면) (ID: banana / PW: 11111)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>** Login Fail **</title>
</head>
<body>
<center>
<h3>** Login Fail **</h3>
<br>
ID, Password가 틀렸습니다.<br>
다시 아래 화면에서 재로그인하세요.
<br>

<jsp:include page="loginForm.jsp"/>

</center>
</body>
</html>
```

