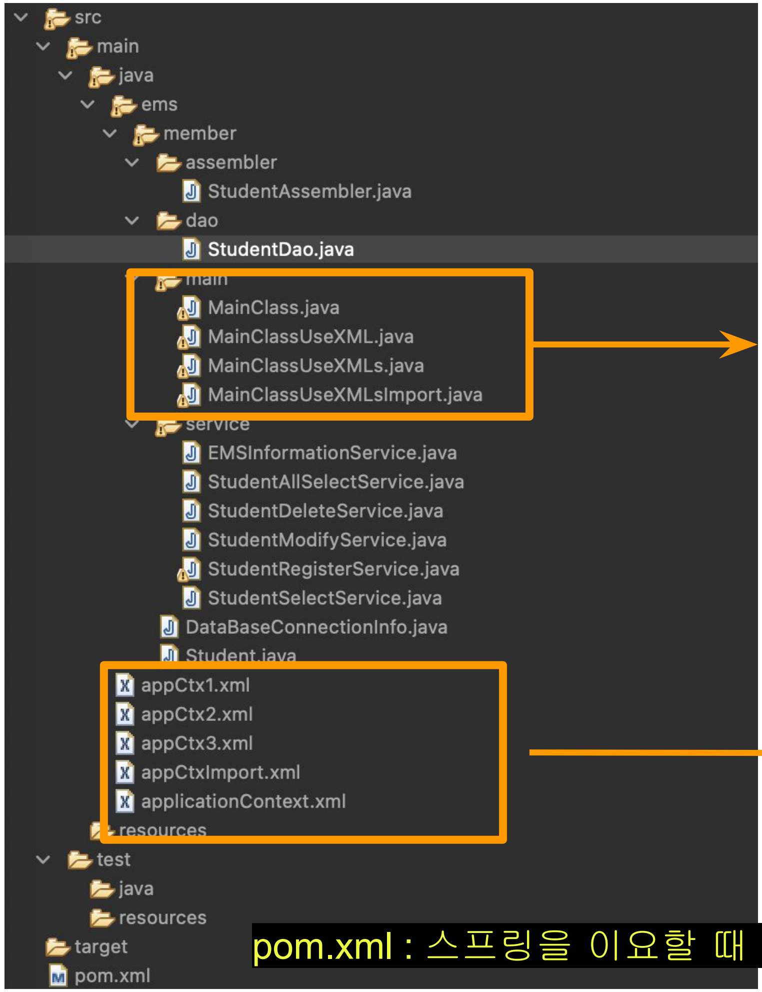
EMS

Education Management System

분석하기. 2023-01-30-월

# 파일 구조



```
src
  main
    java
      ems
        member
          assembler
            StudentAssembler.java
          dao
            StudentDao.java
          main
            MainClass.java
            MainClassUseXML.java
            MainClassUseXMLs.java
            MainClassUseXMLsImport.java
          service
            EMSInformationService.java
            StudentAllSelectService.java
            StudentDeleteService.java
            StudentModifyService.java
            StudentRegisterService.java
            StudentSelectService.java
          DataBaseConnectionInfo.java
          Student.java
      appCtx1.xml
      appCtx2.xml
      appCtx3.xml
      appCtxImport.xml
      applicationContext.xml
    resources
  test
    java
    resources
target
pom.xml
```

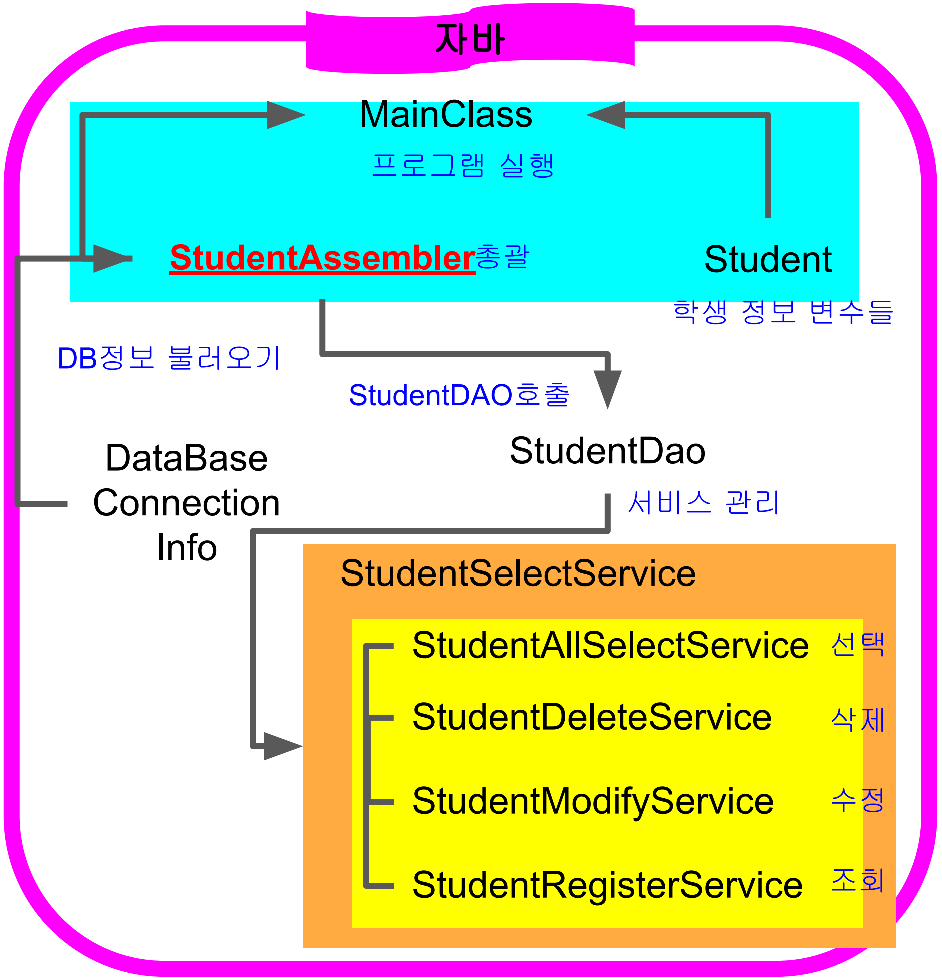
가장 먼저 실행할 수 있는 파일 (여기서 시작)
1. MainClass.java : 기존의 자바 객체 방법을 이용
2. ~~~XML ~~ .java: XML을 이용해 IoC에 DI하는 스프링 방법

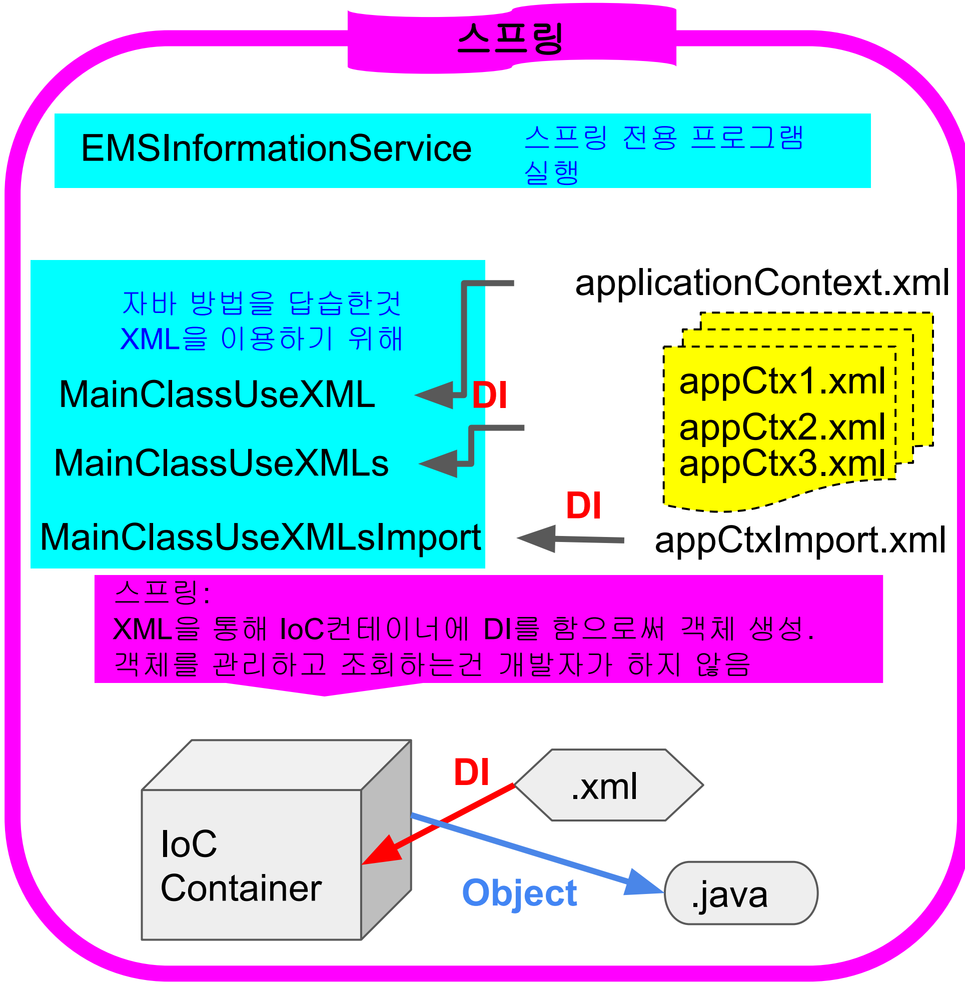
1. EMSInformationService.java: XML방법을 이용할 때 필요

XML을 통해 IoC컨테이너에 DI

pom.xml : 스프링을 이용할 때 기본적으로 필요

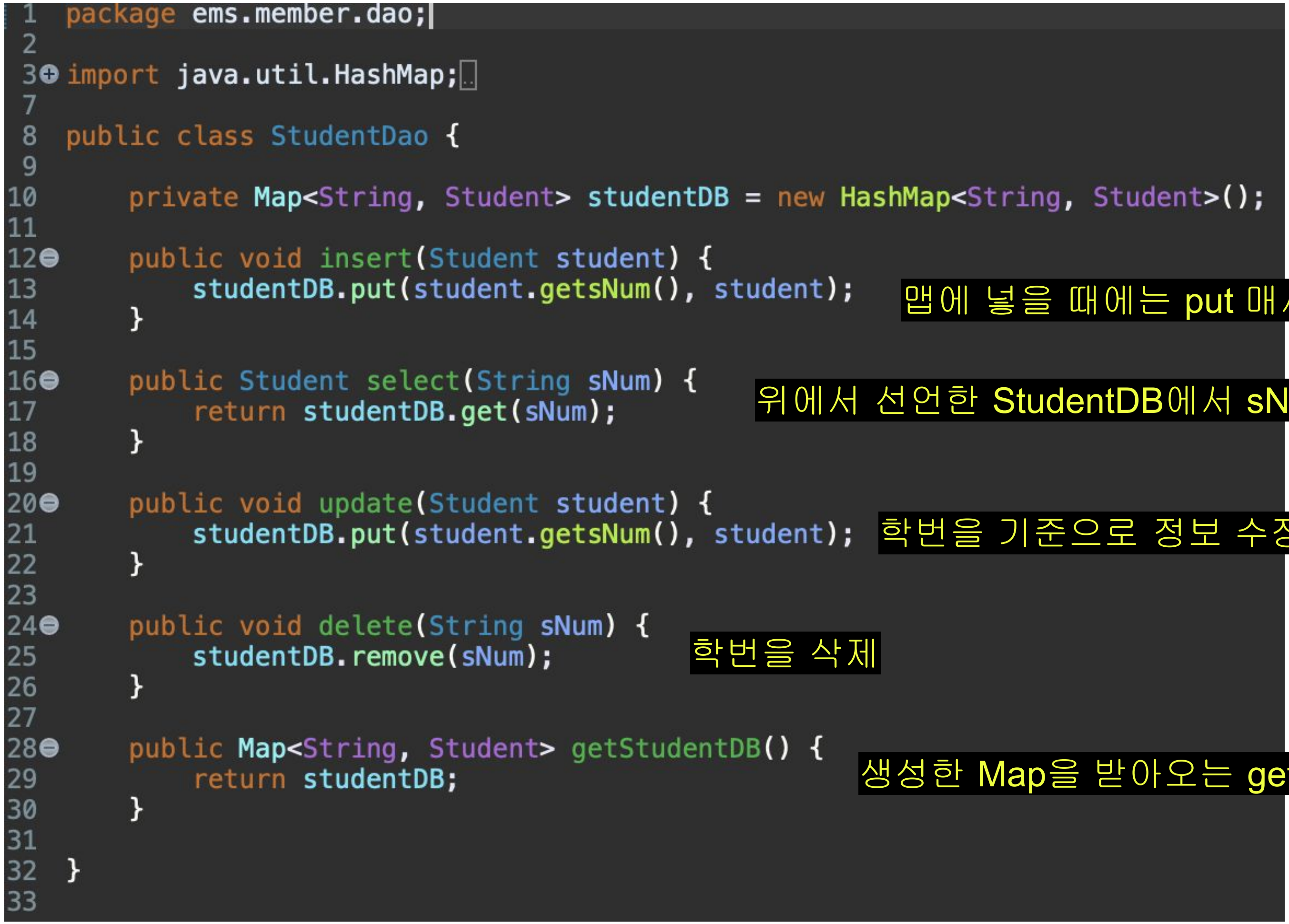
$ 실행 수형도 (.xml제외하고 모두 자바 파일임 ~.java생략)





스프링:
XML을 통해 IoC컨테이너에 DI를 함으로써 객체 생성.
객체를 관리하고 조회하는건 개발자가 하지 않음

StudentDao.java: 기존 자바에서 객체 생성하는 방법으로 구현한 것.



```java
package ems.member.dao;

import java.util.HashMap;

public class StudentDao {

	private Map<String, Student> studentDB = new HashMap<String, Student>();

	public void insert(Student student) {
		studentDB.put(student.getsNum(), student);
	}

	public Student select(String sNum) {
		return studentDB.get(sNum);
	}

	public void update(Student student) {
		studentDB.put(student.getsNum(), student);
	}

	public void delete(String sNum) {
		studentDB.remove(sNum);
	}

	public Map<String, Student> getStudentDB() {
		return studentDB;
	}

}
```

맵 (컬렉션 프레임워크)로 선언

맵에 넣을 때에는 put 매서드를 이용, 학번을 가장 먼저 넣

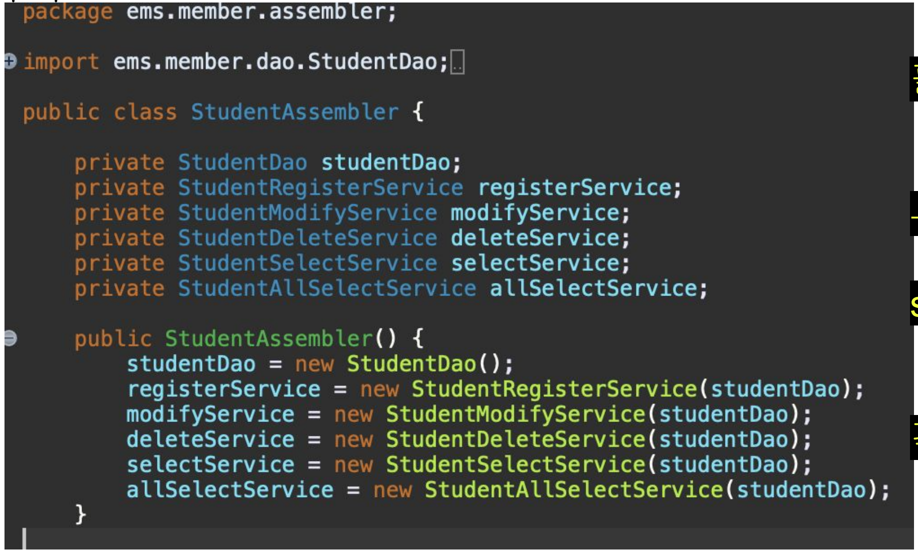
위에서 선언한 StudentDB에서 sNums(학번)을 기준으로 객체 선언

학번을 기준으로 정보 수정

학번을 삭제

생성한 Map을 받아오는 getter

StudentAssembler.java: StudentDao를 통해 다른 패키지들과 연결해주는 컨트롤러 역할을 하는 자바



```
package ems.member.assembler;

import ems.member.dao.StudentDao;

public class StudentAssembler {

    private StudentDao studentDao;
    private StudentRegisterService registerService;
    private StudentModifyService modifyService;
    private StudentDeleteService deleteService;
    private StudentSelectService selectService;
    private StudentAllSelectService allSelectService;

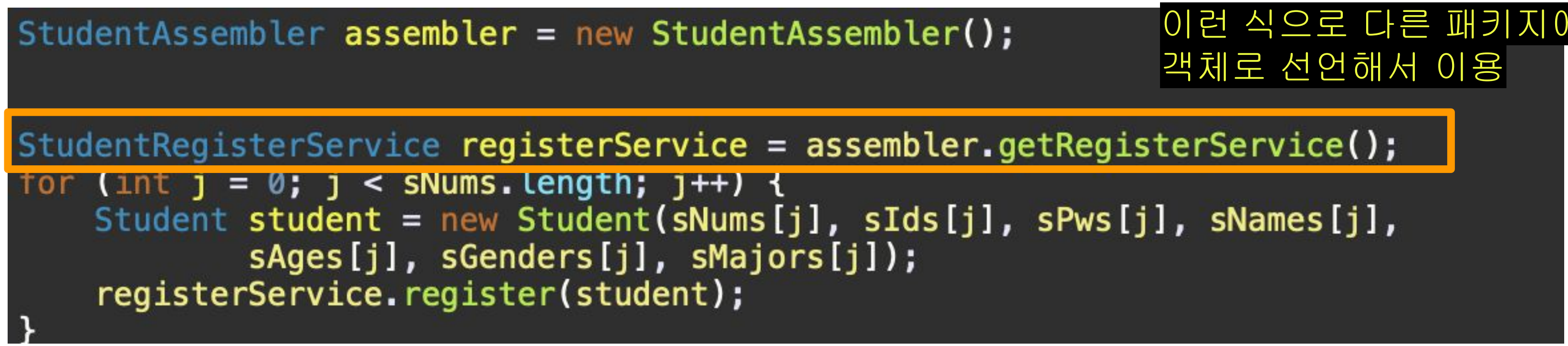
    public StudentAssembler() {
        studentDao = new StudentDao();
        registerService = new StudentRegisterService(studentDao);
        modifyService = new StudentModifyService(studentDao);
        deleteService = new StudentDeleteService(studentDao);
        selectService = new StudentSelectService(studentDao);
        allSelectService = new StudentAllSelectService(studentDao);
    }
```

필요한 자바 파일들을 이용하기 위해 import문에서 모두 불

→ 필드 멤버 변수로 선언

studentDao는 기본적으로 모두 이용.

객체로 선언

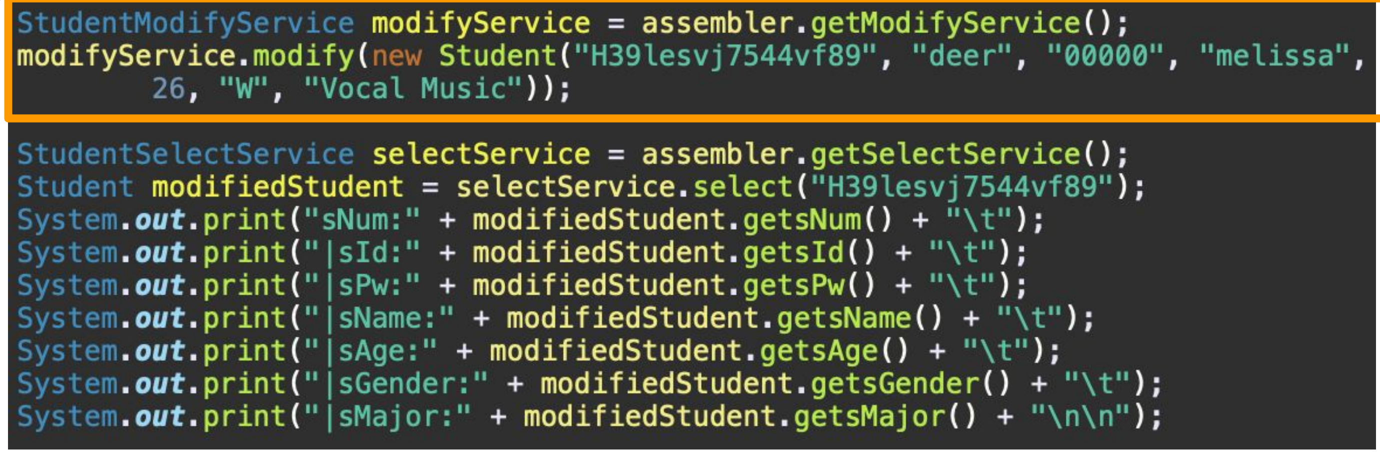


```
StudentAssembler assembler = new StudentAssembler();

StudentRegisterService registerService = assembler.getRegisterService();
for (int j = 0; j < sNums.length; j++) {
    Student student = new Student(sNums[j], sIds[j], sPws[j], sNames[j],
            sAges[j], sGenders[j], sMajors[j]);
    registerService.register(student);
}
```

이런 식으로 다른 패키지에 있는걸
객체로 선언해서 이용

StudentAssembler.java: StudentDao를 통해 다른 패키지들과 연결해주는 컨트롤러 역할을 하는 자바

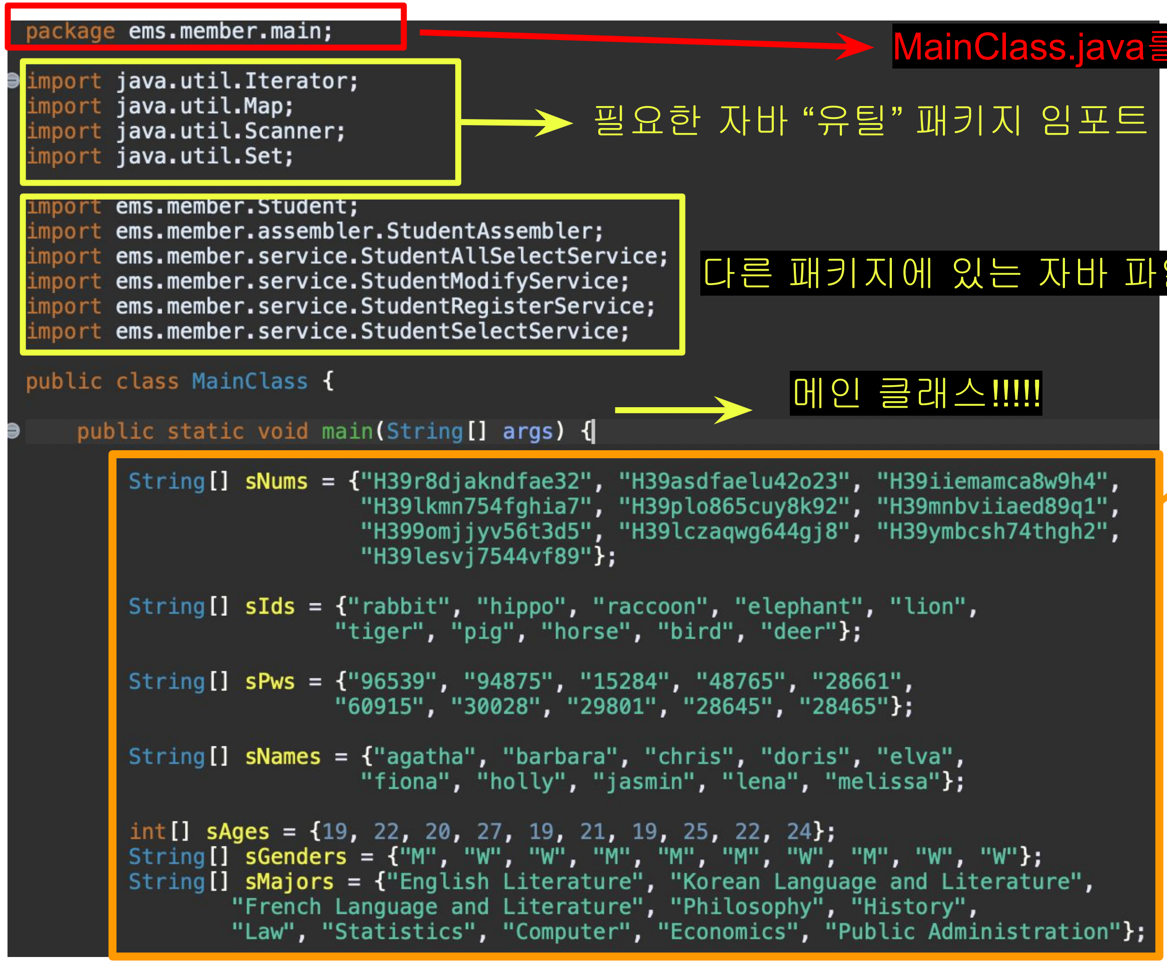


```java
StudentModifyService modifyService = assembler.getModifyService();
modifyService.modify(new Student("H39lesvj7544vf89", "deer", "00000", "melissa",
        26, "W", "Vocal Music"));

StudentSelectService selectService = assembler.getSelectService();
Student modifiedStudent = selectService.select("H39lesvj7544vf89");
System.out.print("sNum:" + modifiedStudent.getsNum() + "\t");
System.out.print("|sId:" + modifiedStudent.getsId() + "\t");
System.out.print("|sPw:" + modifiedStudent.getsPw() + "\t");
System.out.print("|sName:" + modifiedStudent.getsName() + "\t");
System.out.print("|sAge:" + modifiedStudent.getsAge() + "\t");
System.out.print("|sGender:" + modifiedStudent.getsGender() + "\t");
System.out.print("|sMajor:" + modifiedStudent.getsMajor() + "\n\n");
```

modify 매서드를 이용해 임의 학번 수정

MainClass.java: 기존 자바에서 객체 생성하는 방법으로 구현한 것.



```java
package ems.member.main;

import java.util.Iterator;
import java.util.Map;
import java.util.Scanner;
import java.util.Set;

import ems.member.Student;
import ems.member.assembler.StudentAssembler;
import ems.member.service.StudentAllSelectService;
import ems.member.service.StudentModifyService;
import ems.member.service.StudentRegisterService;
import ems.member.service.StudentSelectService;

public class MainClass {

    public static void main(String[] args) {

        String[] sNums = {"H39r8djakndfae32", "H39asdfaelu42o23", "H39iiemamca8w9h4",
                          "H39lkmn754fghia7", "H39plo865cuy8k92", "H39mnbviiaed89q1",
                          "H399omjjyv56t3d5", "H39lczaqwg644gj8", "H39ymbcsh74thgh2",
                          "H39lesvj7544vf89"};

        String[] sIds = {"rabbit", "hippo", "raccoon", "elephant", "lion",
                        "tiger", "pig", "horse", "bird", "deer"};

        String[] sPws = {"96539", "94875", "15284", "48765", "28661",
                        "60915", "30028", "29801", "28645", "28465"};

        String[] sNames = {"agatha", "barbara", "chris", "doris", "elva",
                          "fiona", "holly", "jasmin", "lena", "melissa"};

        int[] sAges = {19, 22, 20, 27, 19, 21, 19, 25, 22, 24};
        String[] sGenders = {"M", "W", "W", "M", "M", "M", "W", "M", "W", "W"};
        String[] sMajors = {"English Literature", "Korean Language and Literature",
                "French Language and Literature", "Philosophy", "History",
                "Law", "Statistics", "Computer", "Economics", "Public Administration"};
```

MainClass.java를 이용하기 위해 가장 먼저 선언하는 코드

필요한 자바 “유틸” 패키지 임포트

다른 패키지에 있는 자바 파일을 이용하기 위해서 임포트

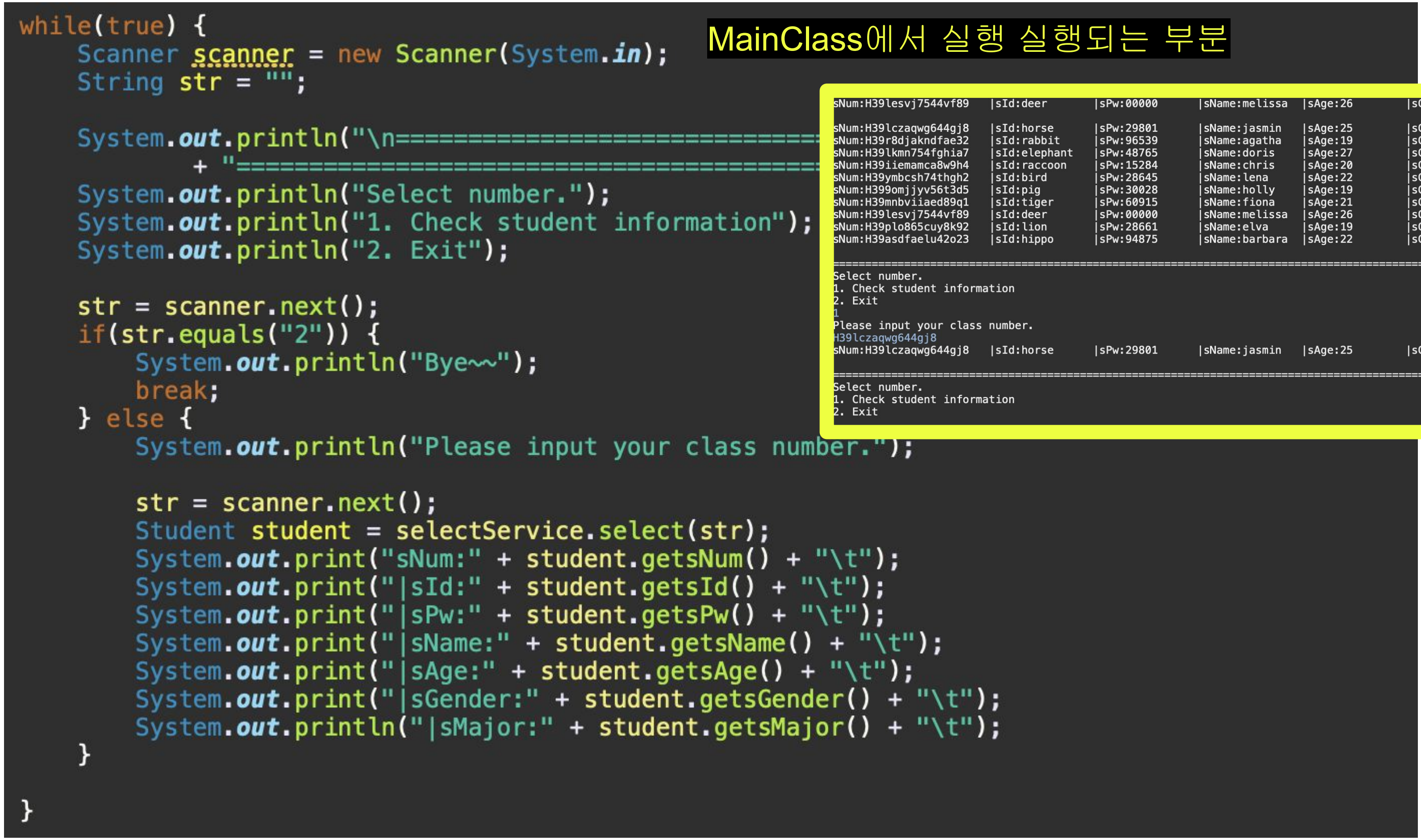
메인 클래스!!!!!

필드 멤버 변수 초기화

sNums	학생 번호(학번)
sIds	학생 아이디(닉네임)
sPws	학생 비밀번호
sNames	학생 이름
sAges	학생 나이
sGenders	학생 성별
sMajors	학생 전공

# MainClass.java: 기존 자바에서 객체 생성하는 방법으로 구현한 것.

MainClass에서 실행 실행되는 부분



```java
while(true) {
    Scanner scanner = new Scanner(System.in);
    String str = "";

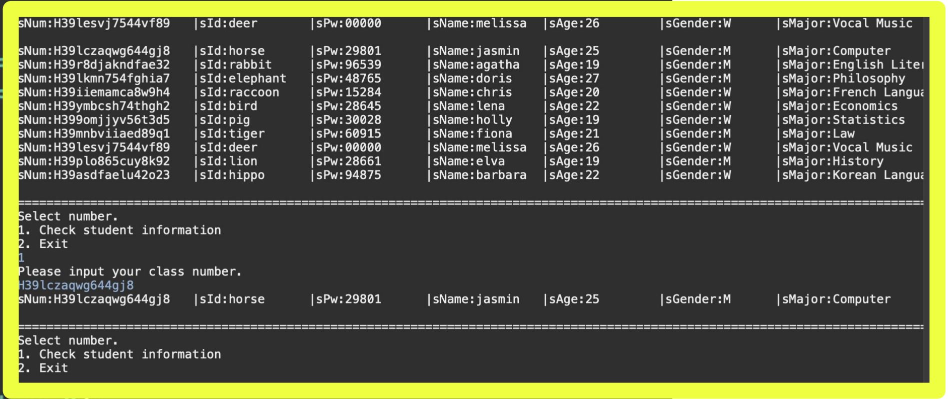
    System.out.println("\n================================
            + "=======================================
    System.out.println("Select number.");
    System.out.println("1. Check student information");
    System.out.println("2. Exit");

    str = scanner.next();
    if(str.equals("2")) {
        System.out.println("Bye~~");
        break;
    } else {
        System.out.println("Please input your class number.");

        str = scanner.next();
        Student student = selectService.select(str);
        System.out.print("sNum:" + student.getsNum() + "\t");
        System.out.print("|sId:" + student.getsId() + "\t");
        System.out.print("|sPw:" + student.getsPw() + "\t");
        System.out.print("|sName:" + student.getsName() + "\t");
        System.out.print("|sAge:" + student.getsAge() + "\t");
        System.out.print("|sGender:" + student.getsGender() + "\t");
        System.out.println("|sMajor:" + student.getsMajor() + "\t");
    }

}
```

실행화면



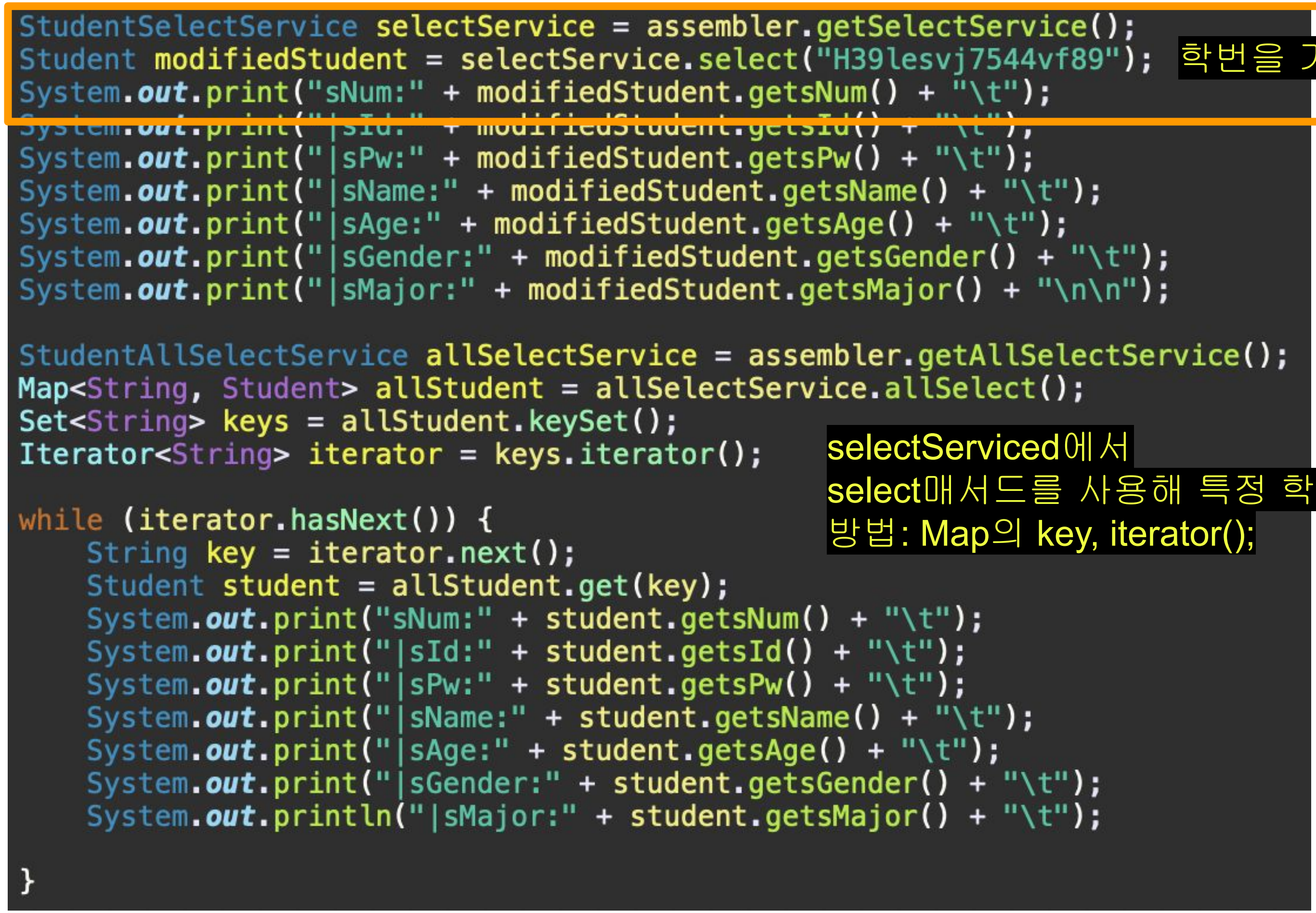
```
sNum:H39lesvj7544vf89   |sId:deer       |sPw:00000      |sName:melissa  |sAge:26        |sGender:W      |sMajor:Vocal Music

sNum:H39lczaqwg644gj8   |sId:horse      |sPw:29801      |sName:jasmin   |sAge:25        |sGender:M      |sMajor:Computer
sNum:H39r8djakndfae32   |sId:rabbit     |sPw:96539      |sName:agatha   |sAge:19        |sGender:M      |sMajor:English Liter
sNum:H39lkmn754fghia7   |sId:elephant   |sPw:48765      |sName:doris    |sAge:27        |sGender:M      |sMajor:Philosophy
sNum:H39iiemamca8w9h4   |sId:raccoon    |sPw:15284      |sName:chris    |sAge:20        |sGender:W      |sMajor:French Langua
sNum:H39ymbcsh74thgh2   |sId:bird       |sPw:28645      |sName:lena     |sAge:22        |sGender:W      |sMajor:Economics
sNum:H399omjjyv56t3d5   |sId:pig        |sPw:30028      |sName:holly    |sAge:19        |sGender:W      |sMajor:Statistics
sNum:H39mnbviiaed89q1   |sId:tiger      |sPw:60915      |sName:fiona    |sAge:21        |sGender:M      |sMajor:Law
sNum:H39lesvj7544vf89   |sId:deer       |sPw:00000      |sName:melissa  |sAge:26        |sGender:W      |sMajor:Vocal Music
sNum:H39plo865cuy8k92   |sId:lion       |sPw:28661      |sName:elva     |sAge:19        |sGender:M      |sMajor:History
sNum:H39asdfaelu42o23   |sId:hippo      |sPw:94875      |sName:barbara  |sAge:22        |sGender:W      |sMajor:Korean Langua

==========================================================================================================================
Select number.
1. Check student information
2. Exit
1
Please input your class number.
H39lczaqwg644gj8
sNum:H39lczaqwg644gj8   |sId:horse      |sPw:29801      |sName:jasmin   |sAge:25        |sGender:M      |sMajor:Computer

==========================================================================================================================
Select number.
1. Check student information
2. Exit
```

MainClass.java: 기존 자바에서 객체 생성하는 방법으로 구현한 것.



```java
StudentSelectService selectService = assembler.getSelectService();
Student modifiedStudent = selectService.select("H39lesvj7544vf89");
System.out.print("sNum:" + modifiedStudent.getsNum() + "\t");
System.out.print("|sId:" + modifiedStudent.getsId() + "\t");
System.out.print("|sPw:" + modifiedStudent.getsPw() + "\t");
System.out.print("|sName:" + modifiedStudent.getsName() + "\t");
System.out.print("|sAge:" + modifiedStudent.getsAge() + "\t");
System.out.print("|sGender:" + modifiedStudent.getsGender() + "\t");
System.out.print("|sMajor:" + modifiedStudent.getsMajor() + "\n\n");

StudentAllSelectService allSelectService = assembler.getAllSelectService();
Map<String, Student> allStudent = allSelectService.allSelect();
Set<String> keys = allStudent.keySet();
Iterator<String> iterator = keys.iterator();

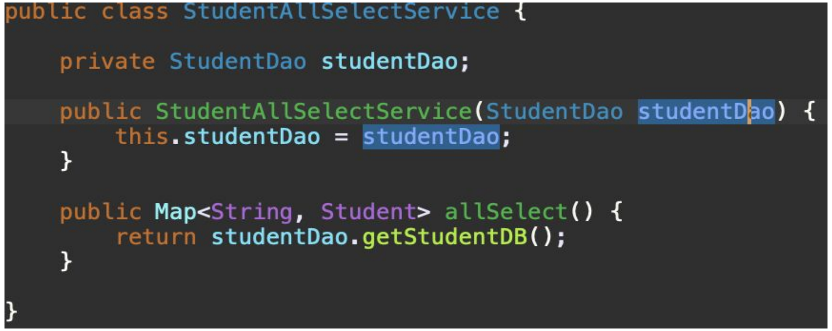
while (iterator.hasNext()) {
    String key = iterator.next();
    Student student = allStudent.get(key);
    System.out.print("sNum:" + student.getsNum() + "\t");
    System.out.print("|sId:" + student.getsId() + "\t");
    System.out.print("|sPw:" + student.getsPw() + "\t");
    System.out.print("|sName:" + student.getsName() + "\t");
    System.out.print("|sAge:" + student.getsAge() + "\t");
    System.out.print("|sGender:" + student.getsGender() + "\t");
    System.out.println("|sMajor:" + student.getsMajor() + "\t");

}
```

학번을 기준으로 수정하는 코드

selectServiced에서
select매서드를 사용해 특정 학번을 불러옴
방법: Map의 key, iterator();

# StudentAllSelectService.java: StudentDao를 이용해 DB에서 불러옴



```
public class StudentAllSelectService {

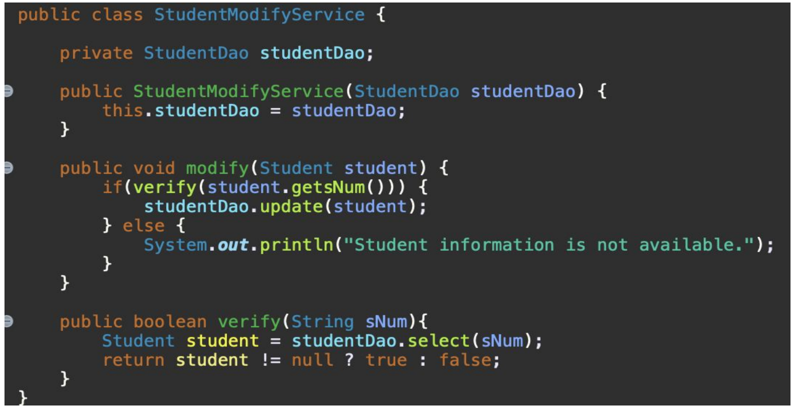
    private StudentDao studentDao;

    public StudentAllSelectService(StudentDao studentDao) {
        this.studentDao = studentDao;
    }

    public Map<String, Student> allSelect() {
        return studentDao.getStudentDB();
    }

}
```

# StudentModifyService.java: 학번을 기준으로 수정



```
public class StudentModifyService {

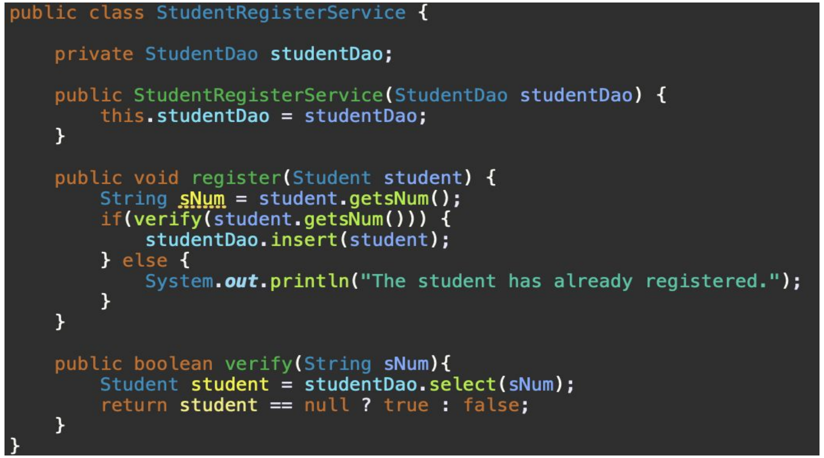
    private StudentDao studentDao;

    public StudentModifyService(StudentDao studentDao) {
        this.studentDao = studentDao;
    }

    public void modify(Student student) {
        if(verify(student.getsNum())) {
            studentDao.update(student);
        } else {
            System.out.println("Student information is not available.");
        }
    }

    public boolean verify(String sNum){
        Student student = studentDao.select(sNum);
        return student != null ? true : false;
    }
}
```

# StudentRegisterService.java: 이미 있는 학생인지 검사



```
public class StudentRegisterService {

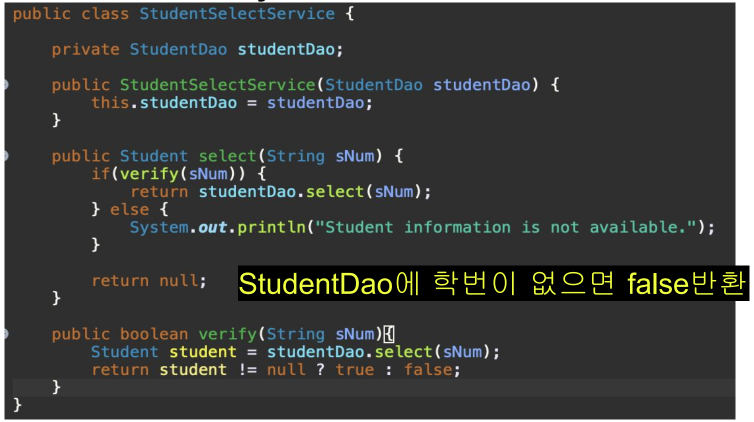
    private StudentDao studentDao;

    public StudentRegisterService(StudentDao studentDao) {
        this.studentDao = studentDao;
    }

    public void register(Student student) {
        String sNum = student.getsNum();
        if(verify(student.getsNum())) {
            studentDao.insert(student);
        } else {
            System.out.println("The student has already registered.");
        }
    }

    public boolean verify(String sNum){
        Student student = studentDao.select(sNum);
        return student == null ? true : false;
    }
}
```

# StudentSelectService.java: 학번을 기준으로 선택



```
public class StudentSelectService {

    private StudentDao studentDao;

    public StudentSelectService(StudentDao studentDao) {
        this.studentDao = studentDao;
    }

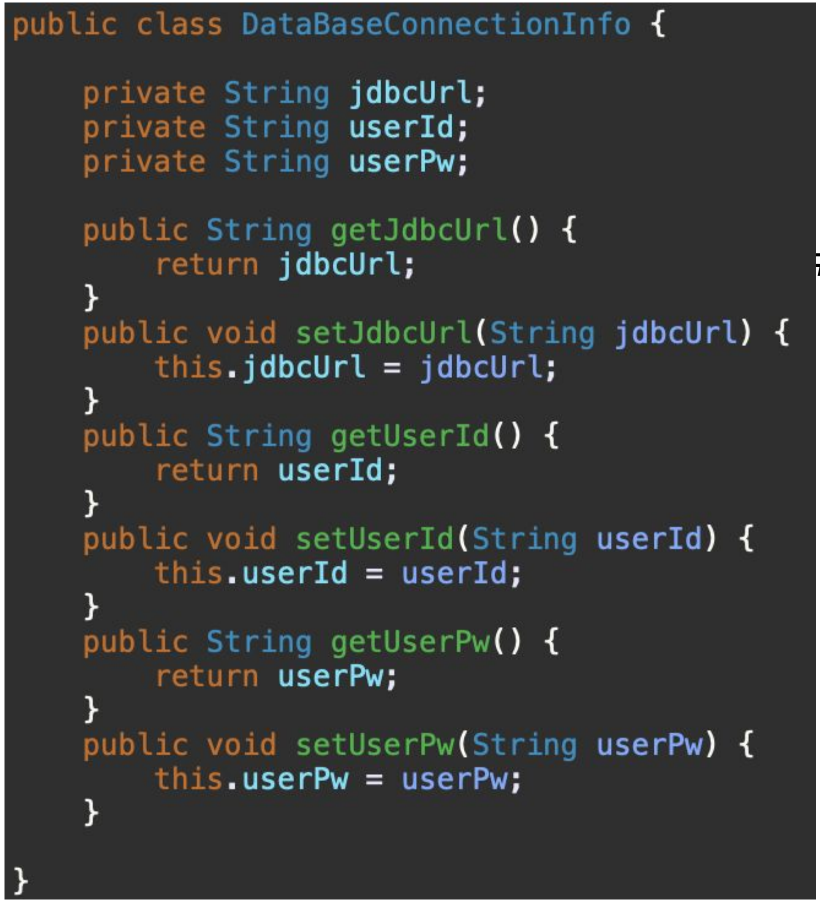
    public Student select(String sNum) {
        if(verify(sNum)) {
            return studentDao.select(sNum);
        } else {
            System.out.println("Student information is not available.");
        }

        return null;
    }

    public boolean verify(String sNum){
        Student student = studentDao.select(sNum);
        return student != null ? true : false;
    }
}
```

StudentDao에 학번이 없으면 false반환

# DataBaseConnectionInfo.java: DB연동을 위한 자바



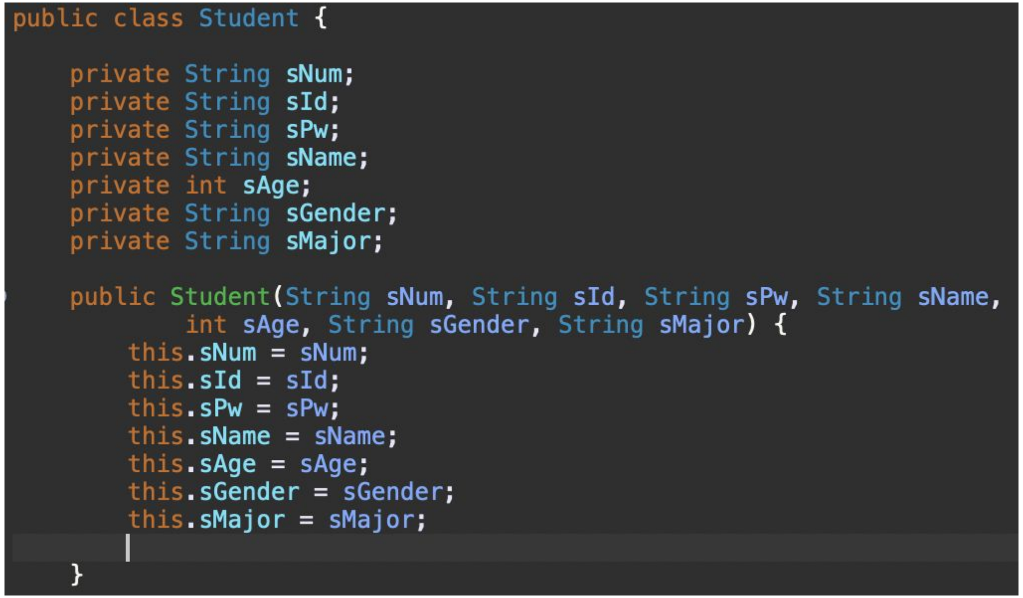
```java
public class DataBaseConnectionInfo {

    private String jdbcUrl;
    private String userId;
    private String userPw;

    public String getJdbcUrl() {
        return jdbcUrl;
    }
    public void setJdbcUrl(String jdbcUrl) {
        this.jdbcUrl = jdbcUrl;
    }
    public String getUserId() {
        return userId;
    }
    public void setUserId(String userId) {
        this.userId = userId;
    }
    public String getUserPw() {
        return userPw;
    }
    public void setUserPw(String userPw) {
        this.userPw = userPw;
    }

}
```

# Student.java: 학생정보를 셋팅하는 자바 (setter, getter)



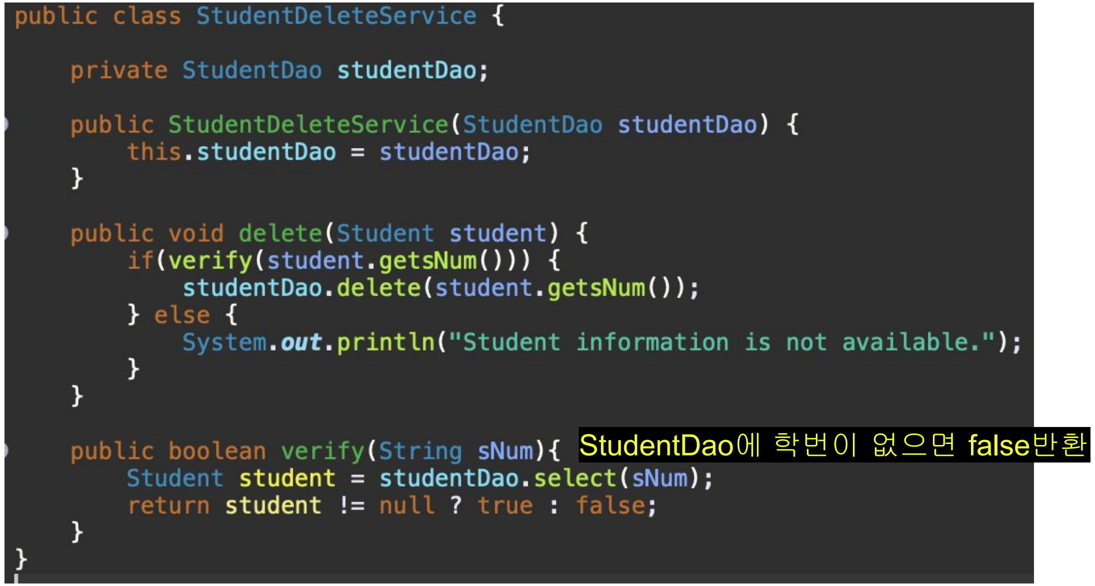
```java
public class Student {

    private String sNum;
    private String sId;
    private String sPw;
    private String sName;
    private int sAge;
    private String sGender;
    private String sMajor;

    public Student(String sNum, String sId, String sPw, String sName,
            int sAge, String sGender, String sMajor) {
        this.sNum = sNum;
        this.sId = sId;
        this.sPw = sPw;
        this.sName = sName;
        this.sAge = sAge;
        this.sGender = sGender;
        this.sMajor = sMajor;

    }
```

StudentDeleteService.java: 학번을 기준으로 삭제하는 자바. → 실행할 때 되지 않음!!??



```
public class StudentDeleteService {

    private StudentDao studentDao;

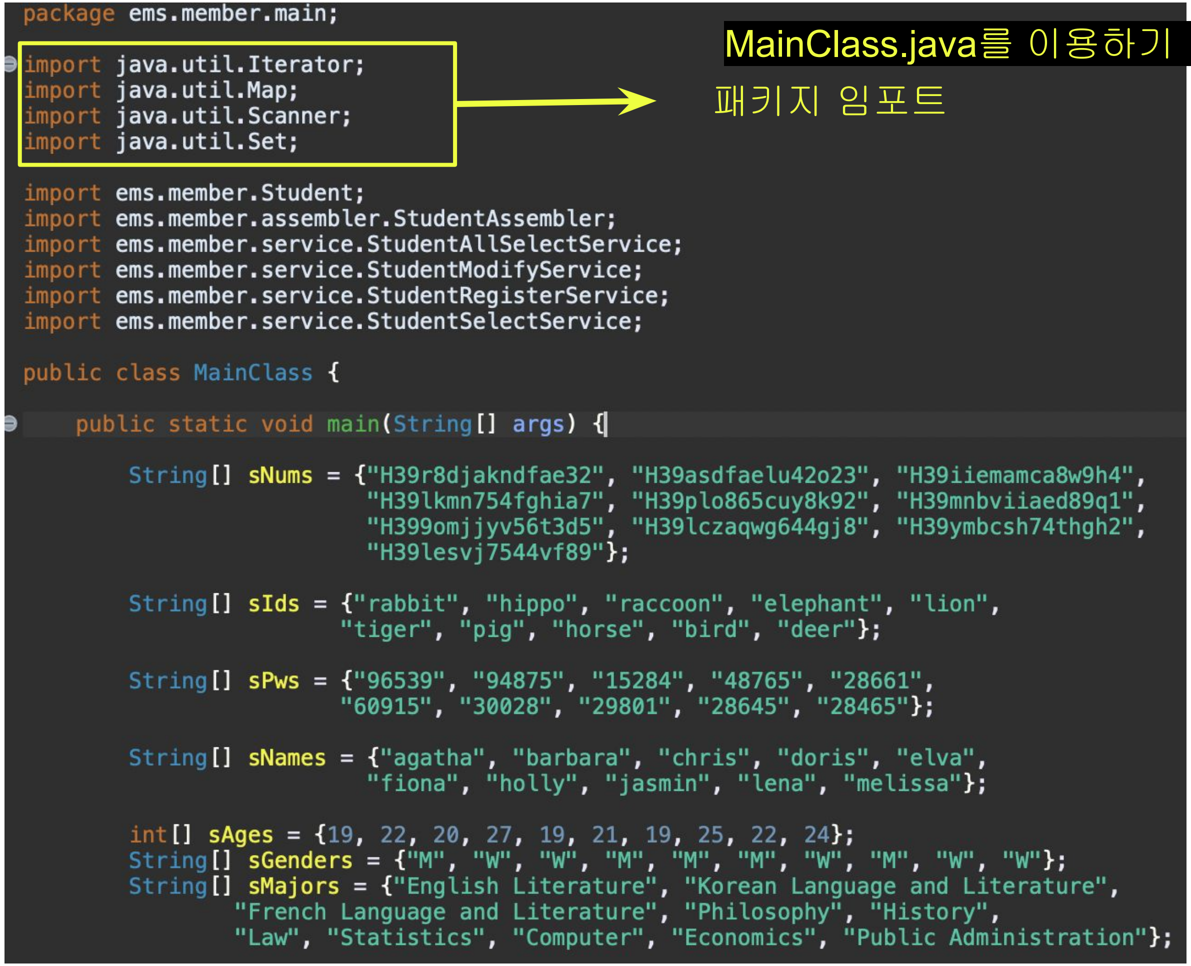
    public StudentDeleteService(StudentDao studentDao) {
        this.studentDao = studentDao;
    }

    public void delete(Student student) {
        if(verify(student.getsNum())) {
            studentDao.delete(student.getsNum());
        } else {
            System.out.println("Student information is not available.");
        }
    }

    public boolean verify(String sNum){
        Student student = studentDao.select(sNum);
        return student != null ? true : false;
    }
}
```

StudentDao에 학번이 없으면 false반환

MainClass.java: 기존 자바에서 객체 생성하는 방법으로 구현한 것.



MainClass.java를 이용하기 위해 가장 먼저 선언하는 코드

패키지 임포트

```java
package ems.member.main;

import java.util.Iterator;
import java.util.Map;
import java.util.Scanner;
import java.util.Set;

import ems.member.Student;
import ems.member.assembler.StudentAssembler;
import ems.member.service.StudentAllSelectService;
import ems.member.service.StudentModifyService;
import ems.member.service.StudentRegisterService;
import ems.member.service.StudentSelectService;

public class MainClass {

    public static void main(String[] args) {

        String[] sNums = {"H39r8djakndfae32", "H39asdfaelu42o23", "H39iiemamca8w9h4",
                          "H39lkmn754fghia7", "H39plo865cuy8k92", "H39mnbviiaed89q1",
                          "H399omjjyv56t3d5", "H39lczaqwg644gj8", "H39ymbcsh74thgh2",
                          "H39lesvj7544vf89"};

        String[] sIds = {"rabbit", "hippo", "raccoon", "elephant", "lion",
                         "tiger", "pig", "horse", "bird", "deer"};

        String[] sPws = {"96539", "94875", "15284", "48765", "28661",
                         "60915", "30028", "29801", "28645", "28465"};

        String[] sNames = {"agatha", "barbara", "chris", "doris", "elva",
                           "fiona", "holly", "jasmin", "lena", "melissa"};

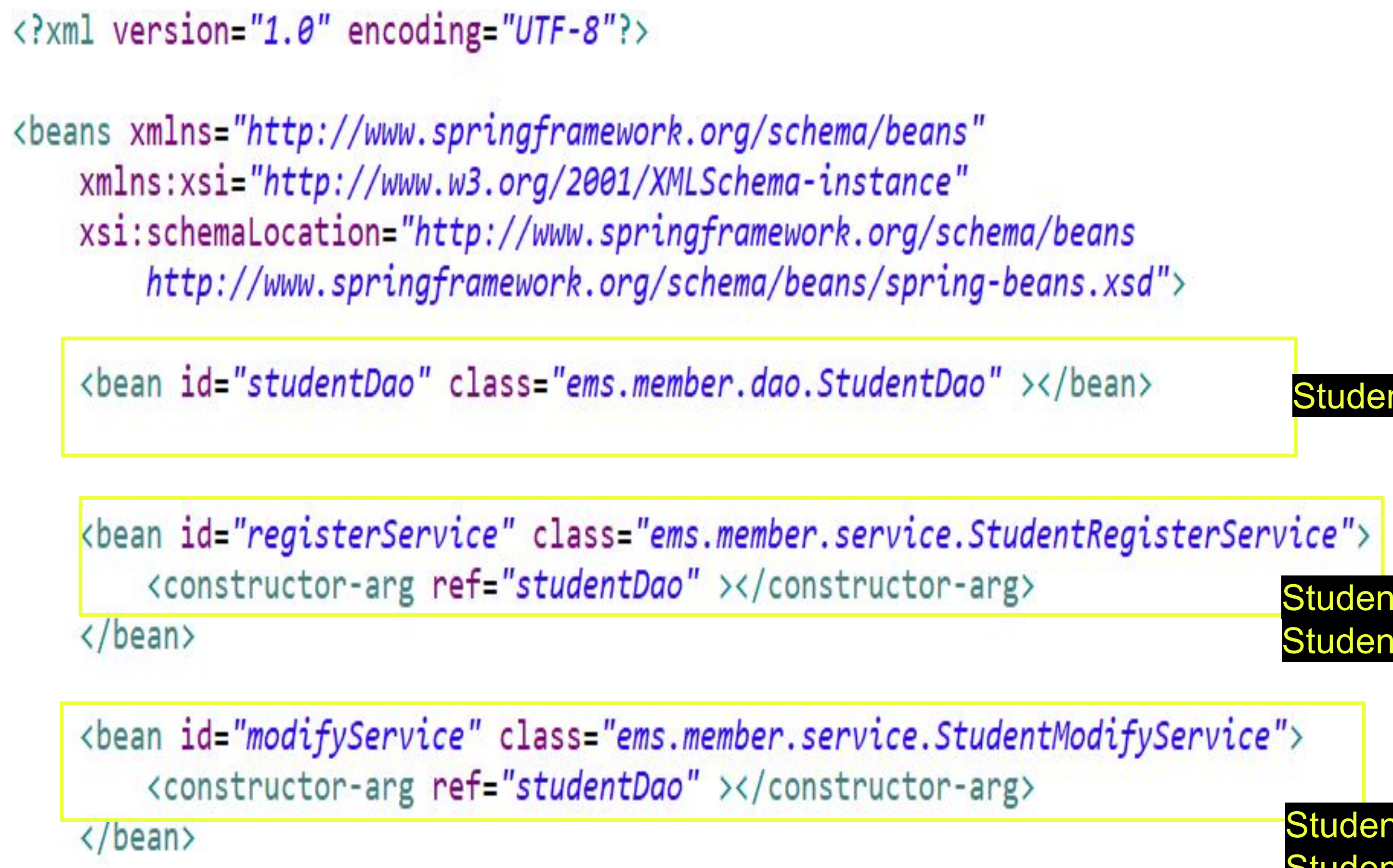
        int[] sAges = {19, 22, 20, 27, 19, 21, 19, 25, 22, 24};
        String[] sGenders = {"M", "W", "W", "M", "M", "M", "W", "M", "W", "W"};
        String[] sMajors = {"English Literature", "Korean Language and Literature",
                "French Language and Literature", "Philosophy", "History",
                "Law", "Statistics", "Computer", "Economics", "Public Administration"};
```

MainClass.java: 기존 자바에서 객체 생성하는 방법으로 구현한 것.

MainClass.java를 이용하기 위해 가장 먼저 선언하는 코드

패키지 임포트

# appCtx1.xml: 생성자를 주입을 통해 bean 객체 생성



```xml
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="studentDao" class="ems.member.dao.StudentDao" ></bean>

    <bean id="registerService" class="ems.member.service.StudentRegisterService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>

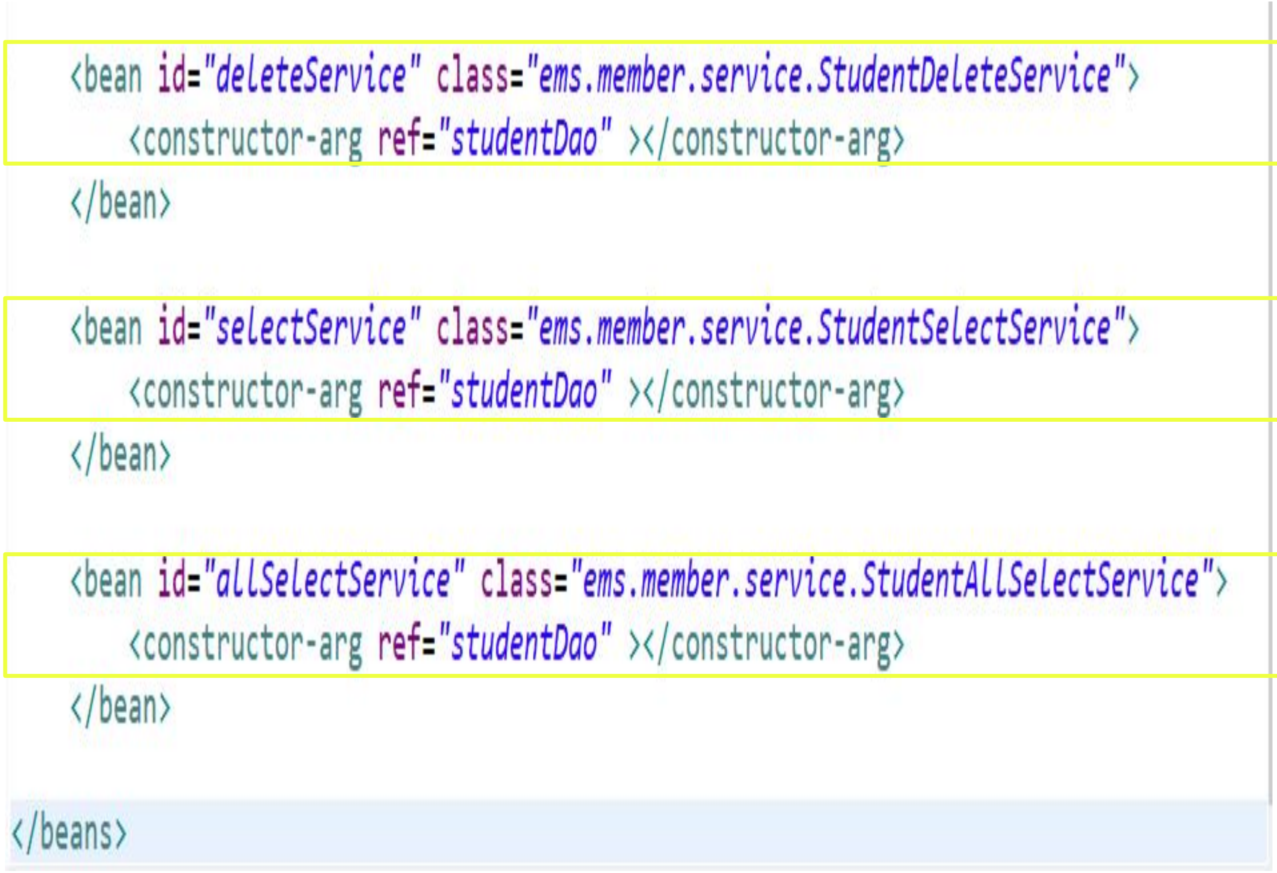
    <bean id="modifyService" class="ems.member.service.StudentModifyService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>
```

StudentDao 빈객체 생성

StudentDao를 참조변수로 받아서
StudentRegisterService 객체생성

StudentDao를 참조변수로 받아서
StudentModifyService 객체생성

# appCtx1.xml: 생성자를 주입을 통해 bean 정의



```
    <bean id="deleteService" class="ems.member.service.StudentDeleteService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>

    <bean id="selectService" class="ems.member.service.StudentSelectService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>

    <bean id="allSelectService" class="ems.member.service.StudentAllSelectService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>

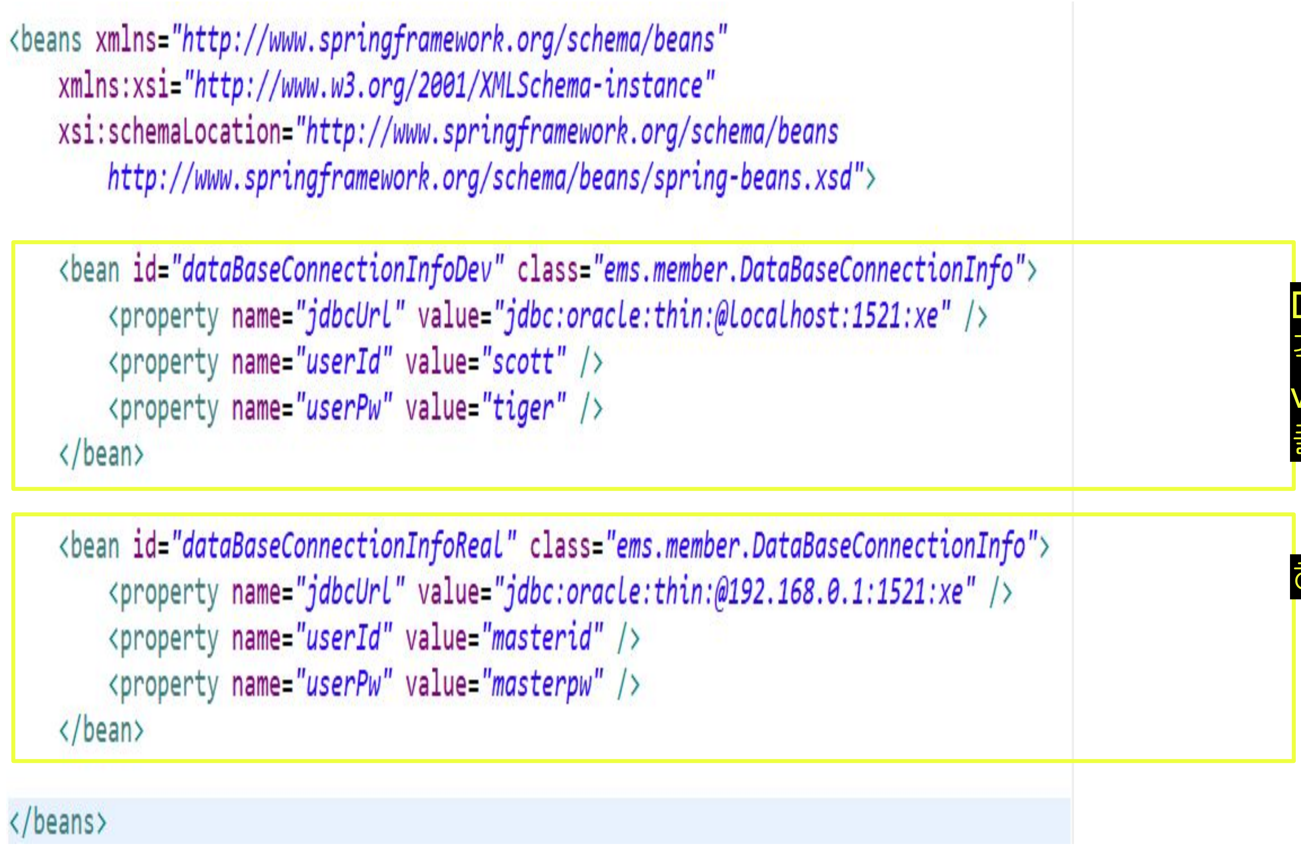
</beans>
```

StudentDao를 참조변수로 받아서
StudentDeleteService 객체생성

StudentDao를 참조변수로 받아서
StudentSelectService 객체생성

StudentDao를 참조변수로 받아서
StudentAllSelectService 객체생성

# appCtx2.xml: setter를 통한 주입으로 필드에 값 저장하고 bean 정의



```
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="dataBaseConnectionInfoDev" class="ems.member.DataBaseConnectionInfo">
        <property name="jdbcUrl" value="jdbc:oracle:thin:@localhost:1521:xe" />
        <property name="userId" value="scott" />
        <property name="userPw" value="tiger" />
    </bean>

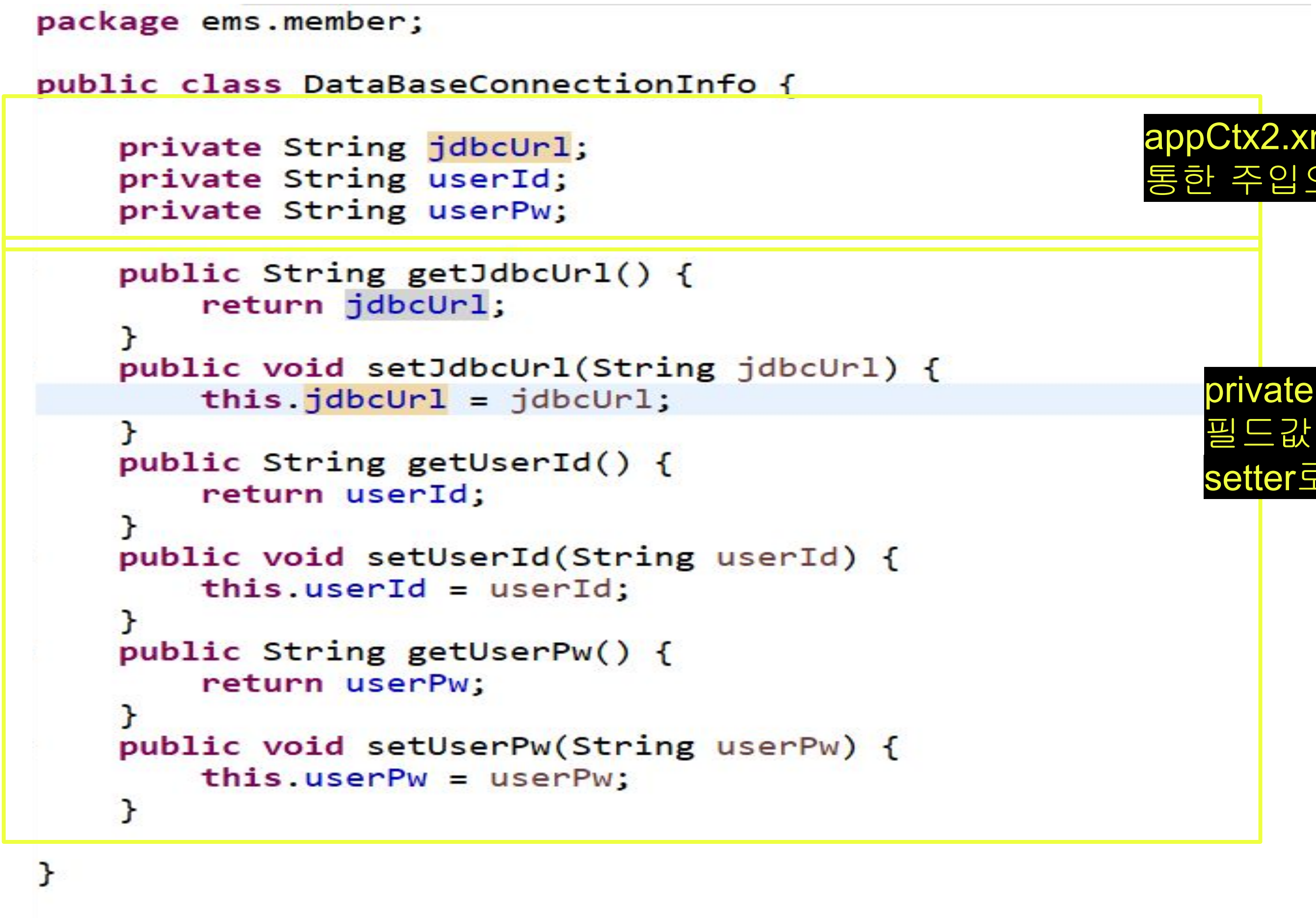
    <bean id="dataBaseConnectionInfoReal" class="ems.member.DataBaseConnectionInfo">
        <property name="jdbcUrl" value="jdbc:oracle:thin:@192.168.0.1:1521:xe" />
        <property name="userId" value="masterid" />
        <property name="userPw" value="masterpw" />
    </bean>

</beans>
```

DataBaseConnectionInfo클래스
객체에서 일치하는 필드값을 찾아
value값을 매칭하여 저장(name값고
클래스의 필드값이 일치해야함)

하나의 클래스에 빈객체를 2개 생

# DataBaseConnectionInfo.java: DB와 연결하는 코드



```java
package ems.member;

public class DataBaseConnectionInfo {

    private String jdbcUrl;
    private String userId;
    private String userPw;

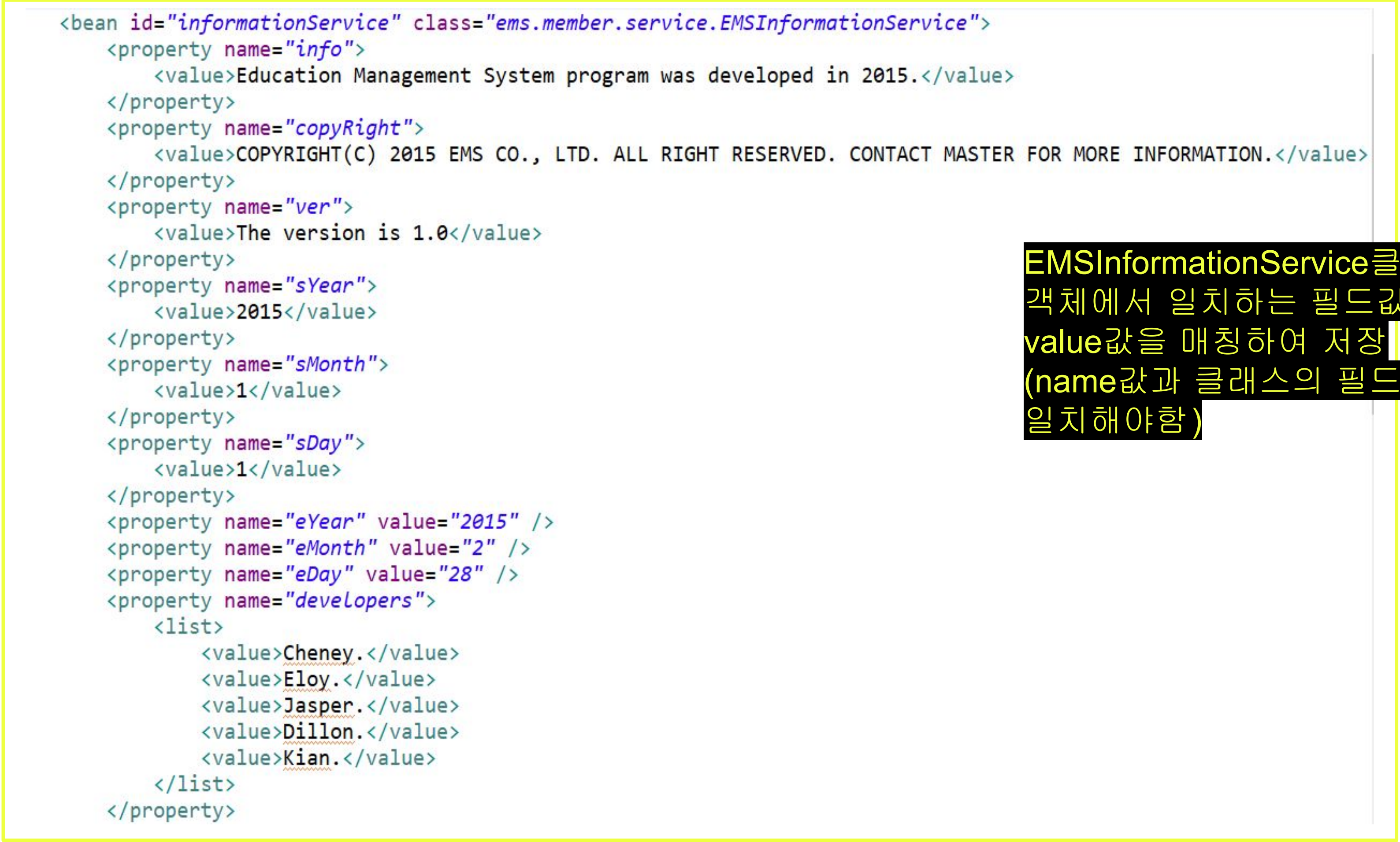
    public String getJdbcUrl() {
        return jdbcUrl;
    }
    public void setJdbcUrl(String jdbcUrl) {
        this.jdbcUrl = jdbcUrl;
    }
    public String getUserId() {
        return userId;
    }
    public void setUserId(String userId) {
        this.userId = userId;
    }
    public String getUserPw() {
        return userPw;
    }
    public void setUserPw(String userPw) {
        this.userPw = userPw;
    }

}
```

appCtx2.xml 파일에서 setter를 통한 주입으로 필드에 값 저장

private 접근제한자로 설정한 필드값에 접근하기 위해 getter, setter로 값을 설정

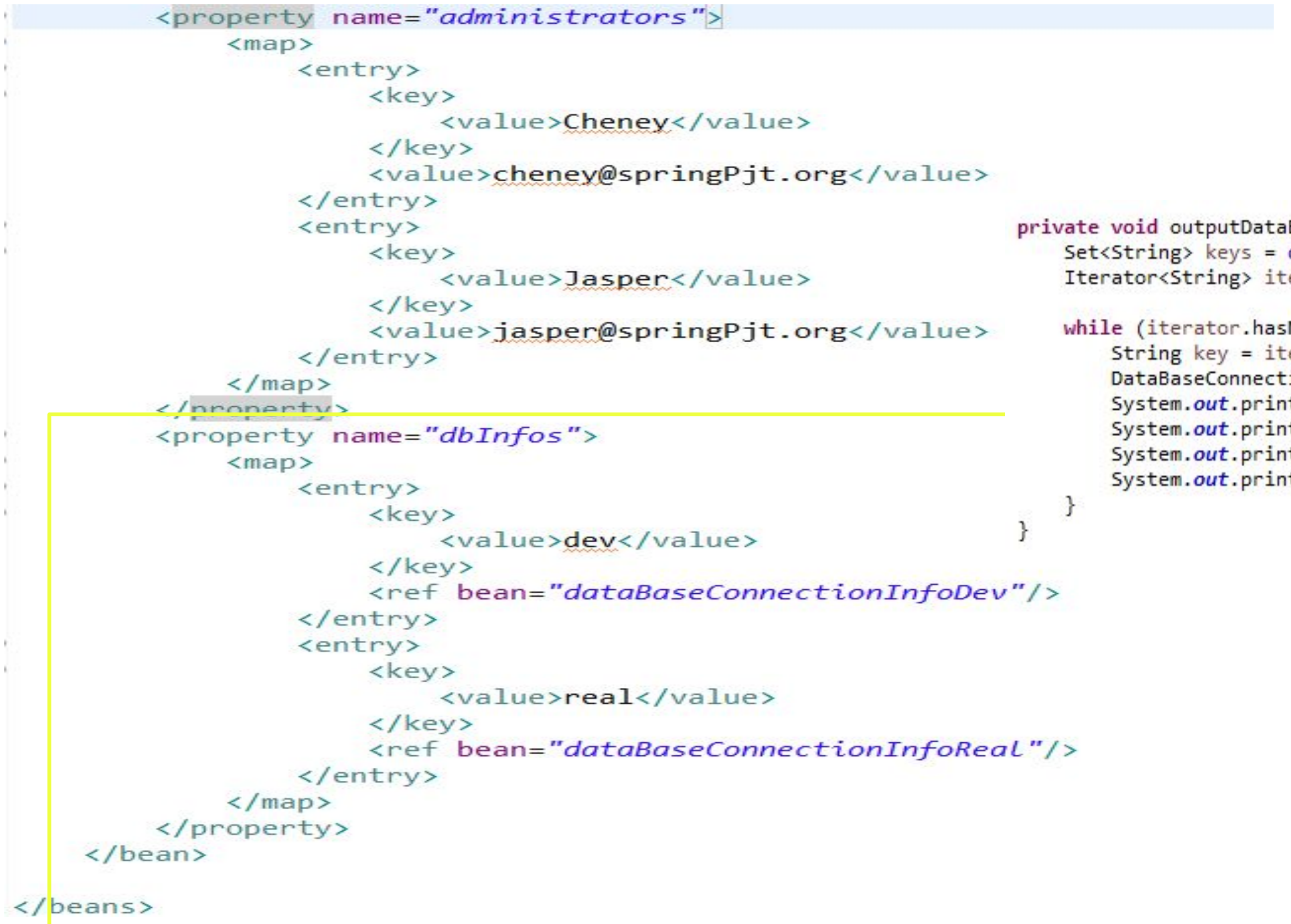
appCtx3.xml: setter를 통한 주입으로 필드값에 값 저장하고 bean 정의



```
<bean id="informationService" class="ems.member.service.EMSInformationService">
    <property name="info">
        <value>Education Management System program was developed in 2015.</value>
    </property>
    <property name="copyRight">
        <value>COPYRIGHT(C) 2015 EMS CO., LTD. ALL RIGHT RESERVED. CONTACT MASTER FOR MORE INFORMATION.</value>
    </property>
    <property name="ver">
        <value>The version is 1.0</value>
    </property>
    <property name="sYear">
        <value>2015</value>
    </property>
    <property name="sMonth">
        <value>1</value>
    </property>
    <property name="sDay">
        <value>1</value>
    </property>
    <property name="eYear" value="2015" />
    <property name="eMonth" value="2" />
    <property name="eDay" value="28" />
    <property name="developers">
        <list>
            <value>Cheney.</value>
            <value>Eloy.</value>
            <value>Jasper.</value>
            <value>Dillon.</value>
            <value>Kian.</value>
        </list>
    </property>
```

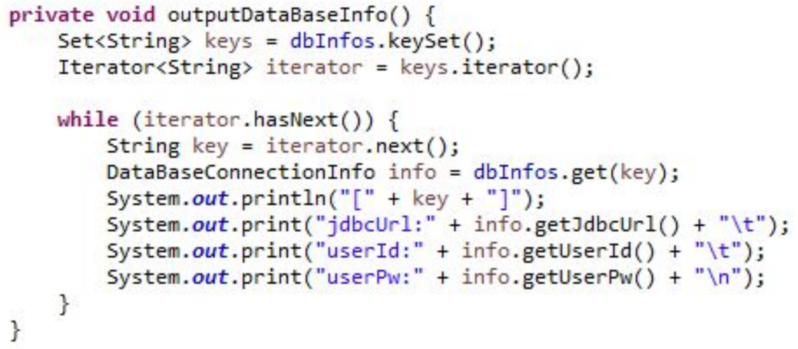
EMSInformationService클래스 객체에서 일치하는 필드값을 찾아 value값을 매칭하여 저장 (name값과 클래스의 필드값이 일치해야함)

# appCtx3.xml: setter를 통한 주입으로 필드값에 값 저장하고 bean 정의



```xml
        <property name="administrators">
            <map>
                <entry>
                    <key>
                        <value>Cheney</value>
                    </key>
                    <value>cheney@springPjt.org</value>
                </entry>
                <entry>
                    <key>
                        <value>Jasper</value>
                    </key>
                    <value>jasper@springPjt.org</value>
                </entry>
            </map>
        </property>
        <property name="dbInfos">
            <map>
                <entry>
                    <key>
                        <value>dev</value>
                    </key>
                    <ref bean="dataBaseConnectionInfoDev"/>
                </entry>
                <entry>
                    <key>
                        <value>real</value>
                    </key>
                    <ref bean="dataBaseConnectionInfoReal"/>
                </entry>
            </map>
        </property>
    </bean>

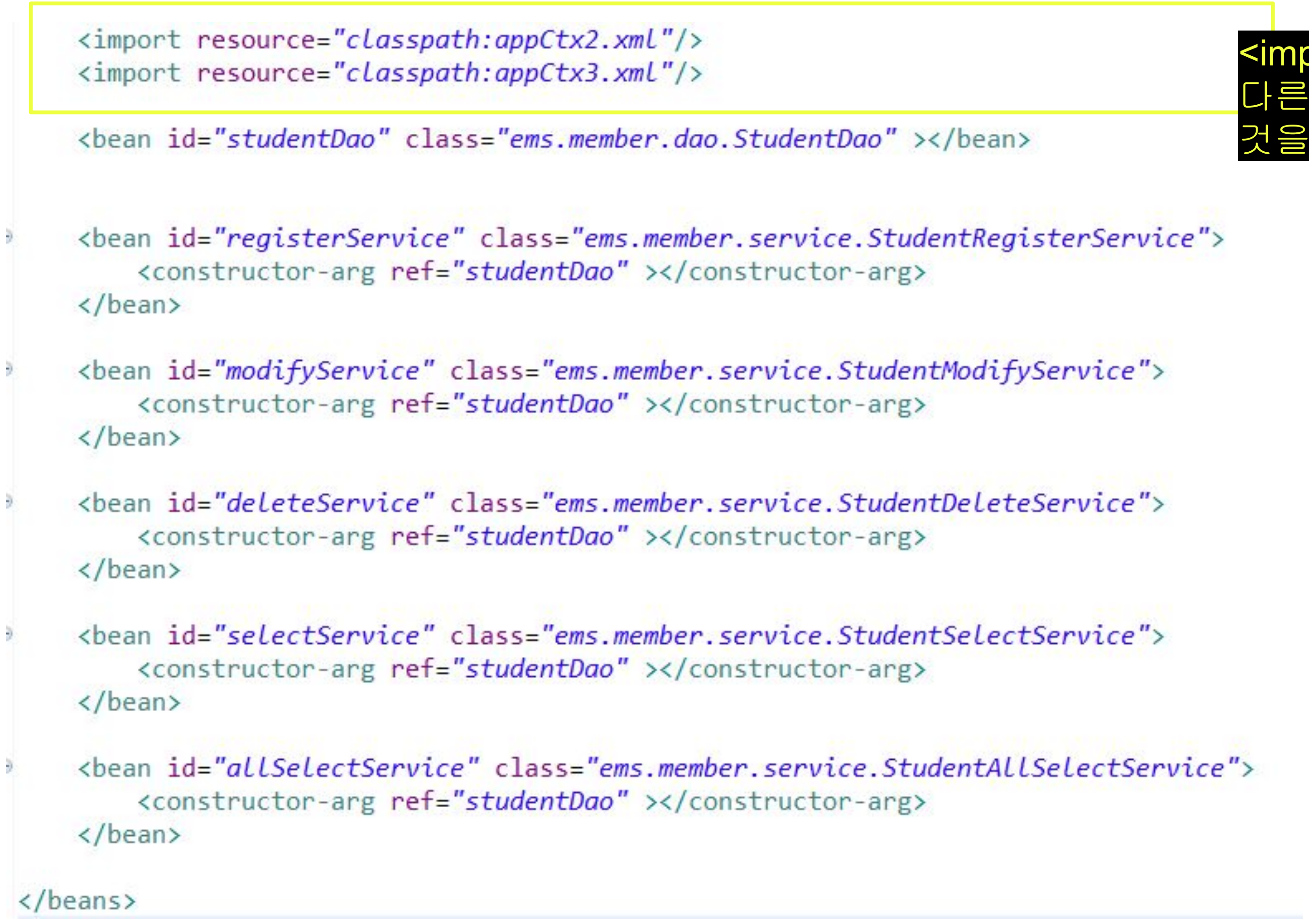
</beans>
```



```java
private void outputDataBaseInfo() {
    Set<String> keys = dbInfos.keySet();
    Iterator<String> iterator = keys.iterator();

    while (iterator.hasNext()) {
        String key = iterator.next();
        DataBaseConnectionInfo info = dbInfos.get(key);
        System.out.println("[" + key + "]");
        System.out.print("jdbcUrl:" + info.getJdbcUrl() + "\t");
        System.out.print("userId:" + info.getUserId() + "\t");
        System.out.print("userPw:" + info.getUserPw() + "\n");
    }
}
```

# appCtxImport.xml: import문으로 경로를 설정해서 자동주입



```xml
    <import resource="classpath:appCtx2.xml"/>
    <import resource="classpath:appCtx3.xml"/>

    <bean id="studentDao" class="ems.member.dao.StudentDao" ></bean>


    <bean id="registerService" class="ems.member.service.StudentRegisterService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>

    <bean id="modifyService" class="ems.member.service.StudentModifyService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>

    <bean id="deleteService" class="ems.member.service.StudentDeleteService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>

    <bean id="selectService" class="ems.member.service.StudentSelectService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>

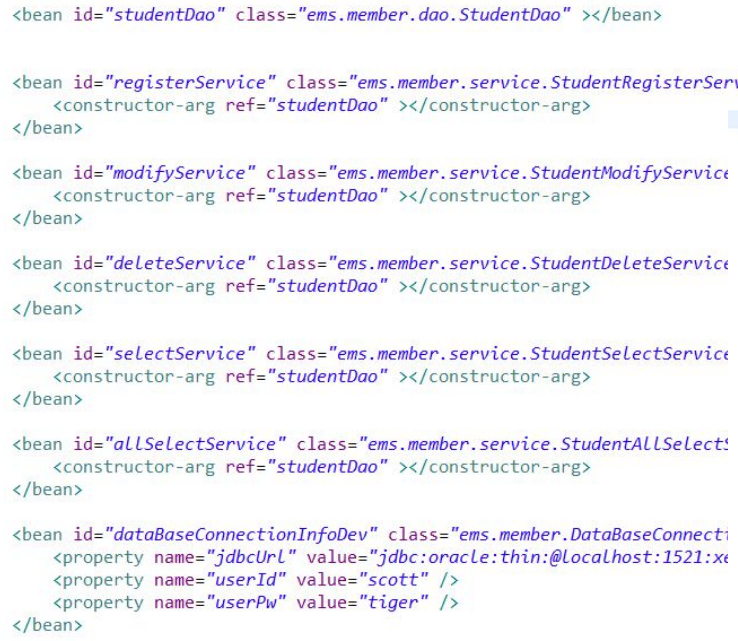
    <bean id="allSelectService" class="ems.member.service.StudentAllSelectService">
        <constructor-arg ref="studentDao" ></constructor-arg>
    </bean>

</beans>
```

<import>태그는 하나의 xml설정에서 다른 xml설정을 함께 사용한다는 것을 지정할 때 사용하는 태그이다.

# applicationContext.xml: 생성자 주입, setter 주입을 이용해서 bean 정의

MainClassUseXml에서 이용



```
<bean id="studentDao" class="ems.member.dao.StudentDao" ></bean>


<bean id="registerService" class="ems.member.service.StudentRegisterService">
    <constructor-arg ref="studentDao" ></constructor-arg>
</bean>

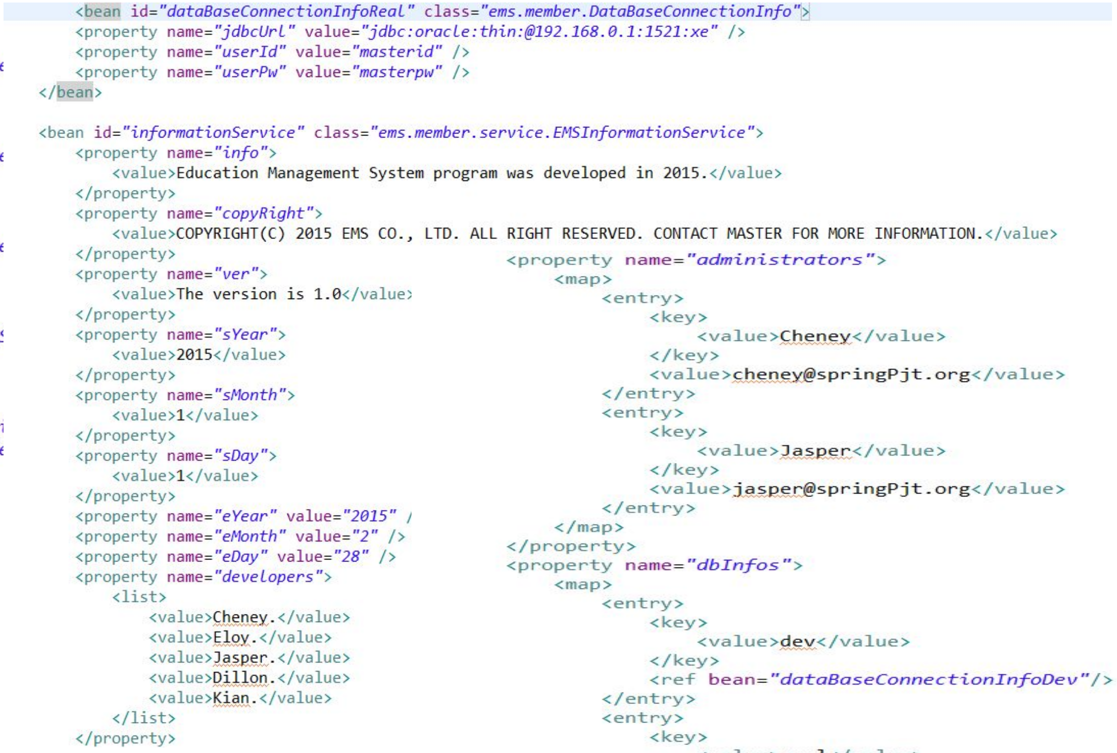
<bean id="modifyService" class="ems.member.service.StudentModifyService
    <constructor-arg ref="studentDao" ></constructor-arg>
</bean>

<bean id="deleteService" class="ems.member.service.StudentDeleteService
    <constructor-arg ref="studentDao" ></constructor-arg>
</bean>

<bean id="selectService" class="ems.member.service.StudentSelectService
    <constructor-arg ref="studentDao" ></constructor-arg>
</bean>

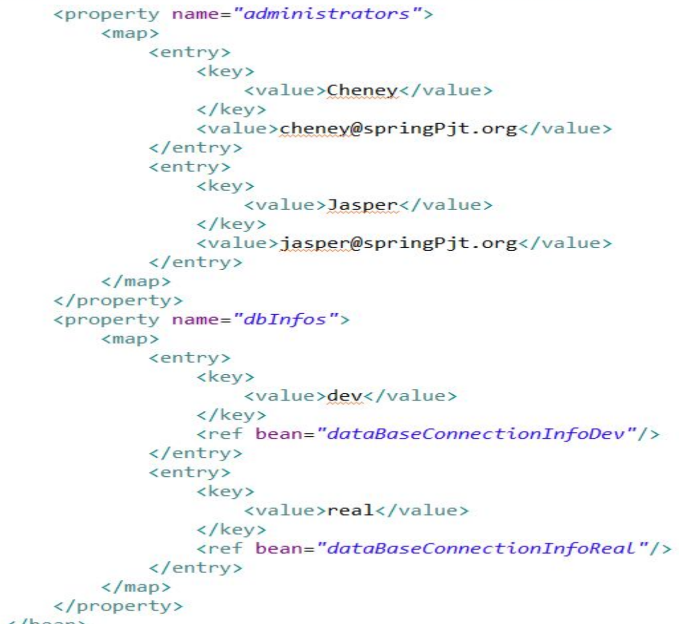
<bean id="allSelectService" class="ems.member.service.StudentAllSelect
    <constructor-arg ref="studentDao" ></constructor-arg>
</bean>

<bean id="dataBaseConnectionInfoDev" class="ems.member.DataBaseConnecti
    <property name="jdbcUrl" value="jdbc:oracle:thin:@localhost:1521:xe
    <property name="userId" value="scott" />
    <property name="userPw" value="tiger" />
</bean>
```



```
    <bean id="dataBaseConnectionInfoReal" class="ems.member.DataBaseConnectionInfo">
    <property name="jdbcUrl" value="jdbc:oracle:thin:@192.168.0.1:1521:xe" />
    <property name="userId" value="masterid" />
    <property name="userPw" value="masterpw" />
</bean>

<bean id="informationService" class="ems.member.service.EMSInformationService">
    <property name="info">
        <value>Education Management System program was developed in 2015.</value>
    </property>
    <property name="copyRight">
        <value>COPYRIGHT(C) 2015 EMS CO., LTD. ALL RIGHT RESERVED. CONTACT MASTER FOR MORE INFORMATION.</value>
    </property>
    <property name="ver">
        <value>The version is 1.0</value>
    </property>
    <property name="sYear">
        <value>2015</value>
    </property>
    <property name="sMonth">
        <value>1</value>
    </property>
    <property name="sDay">
        <value>1</value>
    </property>
    <property name="eYear" value="2015" /
    <property name="eMonth" value="2" />
    <property name="eDay" value="28" />
    <property name="developers">
        <list>
            <value>Cheney.</value>
            <value>Eloy.</value>
            <value>Jasper.</value>
            <value>Dillon.</value>
            <value>Kian.</value>
        </list>
    </property>
```



```
<property name="administrators">
    <map>
        <entry>
            <key>
                <value>Cheney</value>
            </key>
            <value>cheney@springPjt.org</value>
        </entry>
        <entry>
            <key>
                <value>Jasper</value>
            </key>
            <value>jasper@springPjt.org</value>
        </entry>
    </map>
</property>
<property name="dbInfos">
    <map>
        <entry>
            <key>
                <value>dev</value>
            </key>
            <ref bean="dataBaseConnectionInfoDev"/>
        </entry>
        <entry>
            <key>
                <value>real</value>
            </key>
            <ref bean="dataBaseConnectionInfoReal"/>
        </entry>
    </map>
</property>
```