

Міністерство освіти і науки України  
Харківський національний університет імені В.Н. Каразіна  
Факультет комп'ютерних наук

**ЛАБОРАТОРНА РОБОТА № 2**  
з дисципліни «Проектування інформаційних систем»

Тема: «Завдання залежностей між елементами»

Виконали:  
студенти 3 курсу  
групи КС-32  
Бурсак Є.Г.  
Золотухін В.О.  
Перевірив:  
Гамзаєв Р.О.

**Цель работы:** Изучение процесса инициализации бинов в Spring и с технологией разрешения зависимостей между отдельными классами.

**Задание:** Реализовать с использованием технологии Spring взаимодействие с базой данных. Приложение должно быть реализовано в виде консольного и выполнять работу с одной из сущностей: Например Студент и реализовывать операции CRUD.

## ХОД РАБОТЫ

Создаем класс Student который имплементирует Serializable. В данном классе хранятся стандартные сеттеры и геттеры, в которых мы задаем информацию о студентах. Таковую как: Имя, Фамилия, Номер студенческого билета и Факультет.

```
package com.Lab2;

import java.io.Serializable;

public class Student implements Serializable{
    private static final long serialVersionUID = 1L;

    private int ID;
    private String name;
    private String surname;
    private String department;

    Student(){};

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getSurname() {
        return surname;
    }
    public void setSurname(String surname) {
        this.surname = surname;
    }
    public String getDepartment() {
        return department;
    }
    public void setDepartment(String department) {
        this.department = department;
    }
}
```

```

    }

    public int getID() {
        return ID;
    }
    public void setID(int iD) {
        ID = iD;
    }
}

```

Создаем класс Storage, в котором будет создаваться храниться наша база данных. Саму базу данных мы создаем с помощью ArrayList. Методы add() и del() отвечают соответственно за добавление элементов в БД и удаление их из БД. Также в данном классе проводится запись информации из БД в файл и считывание данных с файла.

```

package com.Lab2;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;

public class Storage {
    private String filePath;
    private ArrayList<Student> db;

    Storage(){};

    public ArrayList<Student> getDb() {
        return db;
    }

    public void setDb(ArrayList<Student> db) {
        this.db = db;
    }

    public String getFilePath() {
        return filePath;
    }

    public void setFilePath(String filePath) {
        this.filePath = filePath;
    }

    public void create()
    {
        db = new ArrayList<Student>();
    }

    public void read()
    {
        try {
            ObjectInputStream is = new ObjectInputStream(new
            FileInputStream(filePath));
            db = (ArrayList<Student>)is.readObject();

```

```

        is.close();
    } catch (IOException | ClassNotFoundException ex) {System.out.println("File
Error");}
}

public void add(Student st)
{
    db.add(st);
}

public void del(int pos)
{
    db.remove(pos);
}

public void save()
{
    try {
        ObjectOutputStream os = new ObjectOutputStream(new
FileOutputStream(filePath));
        os.writeObject(db);
        os.close();
    } catch (IOException ex){/*pass*/}
}

}

```

Создаем XML файл в котором происходит регистрация объявленного компонента в контейнере Spring. В данном файле внутри элемента <beans> помещаем полное описание конфигурации Spring, включая объявление <bean>.

```

1<?xml version="1.0" encoding="UTF-8"?>
2<beans xmlns="http://www.springframework.org/schema/beans"
3    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
5    <bean id="Storage" class="com.Lab2.Storage">
6        <property name="filePath" value="db.ser"/>
7    </bean>
8
9</beans>

```

Создаем класс Main, в котором создаем новый объект ApplicationContext. В данный объект мы передаем XML файл, в котором зарегистрированный объявленный компонент в контейнере Spring. Также в данном классе мы реализуем консольный ввод новых элементов и вывод информации из БД на экран.

```

package com.Lab2;

import java.util.ArrayList;
import java.util.Scanner;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {
    public static void main(String[] args) {

```

```

ApplicationContext ctx = new ClassPathXmlApplicationContext(
    "ApplicationResources.xml");
Storage db = (Storage)ctx.getBean("Storage");
Scanner sc = new Scanner(System.in);
boolean flag = true;
while(flag)
{
    switch(menu(sc))
    {
        case 1:
            db.create();
            break;
        case 2:
            db.read();
            break;
        case 3:
            db.add(createStudent(sc));
            break;
        case 4:
            db.del(sc.nextInt());
            break;
        case 5:
            db.save();
            break;
        case 6:
            printDb(db);
            break;
        case 7:
            flag = false;
            break;
        default:
            System.out.println("Invalid value");
            break;
    }
}

}

public static int menu(Scanner sc)
{
    System.out.println("1 Create db");
    System.out.println("2 Read db from file");
    System.out.println("3 Add to db");
    System.out.println("4 Delete from db");
    System.out.println("5 Save to file");
    System.out.println("6 Print db");
    System.out.println("7 Exit");
    System.out.println("Input Value :");

    int read = sc.nextInt();
    return read;
}

public static Student createStudent(Scanner sc)
{
    sc.nextLine();
    Student st = new Student();
    System.out.println("Input ID: ");
    st.setID(Integer.parseInt(sc.nextLine()));
}

```

```

        System.out.println("Input name: ");
        st.setName(sc.nextLine());
        System.out.println("Input surname: ");
        st.setSurname(sc.nextLine());
        System.out.println("Input department: ");
        st.setDepartment(sc.nextLine());
        return st;
    }
    public static void printDb(Storage db)
    {
        ArrayList <Student> store = db.getDb();
        for(int i=0;i<store.size();i+=1)
        {
            System.out.println("Student #" + i);
            System.out.println("ID: " + store.get(i).getID());
            System.out.println("Name: " + store.get(i).getName());
            System.out.println("Surname: " + store.get(i).getSurname());
            System.out.println("Department: " + store.get(i).getDepartment());
            System.out.println();
        }
    }
}

```

## ВЫВОДЫ

В процессе выполнения данной лабораторной работы мы создали базу данных. Студент и изучили процесс инициализации бинов в Spring и с технологией разрешения зависимостей между отдельными классами.

Также мы реализовали взаимодействие с базой данных с использованием технологии Spring.