

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования



НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий
Кафедра информатики и систем управления

Отчет

по лабораторной работе №2

по дисциплине

Шаблоны проектирования программного обеспечения

РУКОВОДИТЕЛЬ:

(подпись)

Жевнерчук Д.В.
(фамилия, и.,о.)

СТУДЕНТ:

Соляник Д

(подпись)

Куприхин Д
(фамилия, и.,о.)

19-ИВТ-2
(шифр группы)

Работа защищена «__» _____

С оценкой _____

Нижний Новгород 2021

Вариант 17

Разработайте и реализуйте объектно-ориентированную модель, на основе которой можно будет создать автоматизированную систему регистрации и сопровождения индивидуальных траекторий обучения. Индивидуальная траектория обучения складывается из:

- обязательных для всех дисциплин;
- рекомендуемых факультативных дисциплин, из которых учащимся может быть выбрано не менее n дисциплин.

В рамках обязательных и факультативных дисциплин могут выполняться учебные проекты, которые представляют собой множество задач, причем каждая задача может быть зарегистрирована в качестве альтернативного варианта лабораторной или практической работы.

Ход работы

В данной лабораторной работе мы использовали Spring Framework для создания и связи объектов. В ходе лабораторной работы создали контекст приложения из xml файла:

```
ClassPathXmlApplicationContext ctx = new  
ClassPathXmlApplicationContext("appAutoscanContext.xml");
```

Создаем с его помощью бины:

```
Obligatory obligatory = ctx.getBean("ob",Obligatory.class);  
Facultative facultative = ctx.getBean("fa",Facultative.class);  
Raspisanie araspisanie = ctx.getBean("ra",Raspisanie.class);
```

Предварительно расставив “маркеры” над необходимыми классами:

```
@Component("fa")  
@Scope("singleton")  
public class Facultative{  
  
@Component("ob")  
@Scope("singleton")  
@PropertySource("topicpart.properties")  
public class Obligatory{  
  
@Component("ra")  
@Scope("singleton ")  
public class Raspisanie{  
  
ctx.close();
```

Результаты:

```
15:25:12.802 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean 'fa'
15:25:12.912 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean 'ob'
15:25:12.920 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean 'ra'
```

Для заполнения объектов информацией использовали Spring. Указывали файл из которого будем считывать данные

`@PropertySource("topicpart.properties")` и значения, которые необходимо считать `@Value("${facultative.name}")` .

Основная суть алгоритма и методы корректировки расписания практически не изменились. Подводя итоги, по сравнению с первой лабораторной работы было сильно уменьшено количество хард-кода, благодаря чему программа приняла более аккуратный и опрятный вид, изменен способ создания и взаимоотношения объектов.

Код программы

App.java

```
package main;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import springhw.scanbeans.Raspisanie;
import springhw.scanbeans.Facultative;
import springhw.scanbeans.Obligatory;

public class App {

    public static void main(String[] args) {
        ClassPathXmlApplicationContext ctx = new
ClassPathXmlApplicationContext("appAutoscanContext.xml");

        System.out.println("Обязательное расписание для Вас:");
        Obligatory obligatory = ctx.getBean("ob",Obligatory.class);
        obligatory.printer();
        Facultative facultative = ctx.getBean("fa",Facultative.class);
        Raspisanie araspisanie = ctx.getBean("ra",Raspisanie.class);
        araspisanie.facultativeRaspisanie(facultative,obligatory);
        System.out.println("Окончательное расписание для Вас:");
        obligatory.printer();
        araspisanie.printFacultative(facultative);
        ctx.close();
    }
}
```

Facultative.java

```
package springhw.scanbeans;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component("fa")
@Scope("singleton")
public class Facultative{
    private String[] info;
    private String[] name;
    private String[] id;
    private String[] object;
    private String[] name_discipline;
    private int[] kolvo_lab = new int[]{ 1,0,1,0};
    private int[] kolvo_pract = new int[]{ 0,1,0,1};
    private int[] kolvo_discipline = new int[] {2,2};

    public int getKolvoObjects() {
        return name_discipline.length;
    }

    public String[] getName() {
        return name;
    }

    public int getKolvoId() {
        return id.length;
    }

    public String[] getId() {
        return id;
    }

    public String[] getInfo() {
        return info;
    }

    public String[] getObject() {
        return object;
    }

    public int[] getKolvoLab() {
        return kolvo_lab;
    }

    public int[] getKolvoPract() {
        return kolvo_pract;
    }

    public String[] getNameDiscipline() {
        return name_discipline;
    }
}
```

```

    public int[] getKolvoDiscipline() {
        return kolvo_discipline;
    }

    @Value("#{${facultative.name}'.split(',')}")
    public void setName(String[] name) {
        this.name = name;
    }

    @Value("#{${facultative.main.name}'.split(',')}")
    public void setNameDiscipline(String[] name_discipline) {
        this.name_discipline = name_discipline;
    }

    @Value("#{${facultative.id}'.split(',')}")
    public void setId(String[] id) {
        this.id = id;
    }

    @Value("#{${facultative.info}'.split(',')}")
    public void setInfo(String[] info) {
        this.info = info;
    }

    @Value("#{${facultative.object}'.split(',')}")
    public void setObject(String[] object) {
        this.object = object;
    }

    // @Value("${facultative.lab}")
    // public void setKolvoLab(ArrayList<Integer> kolvo_lab) {
    //     this.kolvo_lab = kolvo_lab;
    // }
    //
    // @Value("${facultative.pract}")
    // public void setKolvoPract(ArrayList<Integer> kolvo_pract) {
    //     this.kolvo_pract = kolvo_pract;
    // }
}

```

Obligatory.java

```

package springhw.scanbeans;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.PropertySource;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component("ob")
@Scope("singleton")
@PropertySource("topicpart.properties")

```

```

public class Obligatory{
    private String[] name;
    private String[] lab_topic;
    private String[] pract_topic;
    int[] kolvo_lab = new int[]{4,4};
    int[] kolvo_pract = new int[]{5,4};

    public String[] getName() {
        return name;
    }
    public String[] getLab_topic() {
        return lab_topic;
    }

    public int[] getKolvo_lab() {
        return kolvo_lab;
    }

    public int[] getKolvo_pract() {
        return kolvo_pract;
    }

    public String[] getPract_topic() {
        return pract_topic;
    }

    @Value("#{${obligatory.lab_topic}.split(',')}")
    // @Value("${tp.pract_topic}")
    public void setLabTopic(String[] lab_topic) {
        this.lab_topic = lab_topic;
    }

    @Value("#{${obligatory.pract_topic}.split(',')}")
    public void setPractTopic(String[] pract_topic) {
        this.pract_topic = pract_topic;
    }

    @Value("#{${obligatory.name}.split(',')}")
    public void setName(String[] name) {
        this.name = name;
    }

    public void setKolvoLab(int i, int kolvo_lab) {
        this.kolvo_lab[i] = this.kolvo_lab[i] - kolvo_lab;
    }

    public void setKolvoPract(int i, int kolvo_pract) {
        this.kolvo_pract[i] = this.kolvo_pract[i] - kolvo_pract;
    }

    public void printer() {
        int g = 0;
        int h = 0;
        for (int i = 0; i < name.length; i++) {

```

```

        System.out.println(" По предмету " + name[i] + " у вас
запланировано:\n** " + kolvo_lab[i] + " лабораторных работ:");
        for (int j = 0; j < kolvo_lab[i]; j++) {
            System.out.println(lab_topic[g]);
            g++;
        }
        System.out.println("\n** " + kolvo_pract[i] + "
практических работ:");
        for (int j = 0; j < kolvo_pract[i]; j++) {
            System.out.println(pract_topic[h]);
            h++;
        }
    }
}
}
}

```

Raspisanie.java

```

package springhw.scanbeans;

import java.util.Scanner;

import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component("ra")
@Scope("singleton")
public class Raspisanie{
    private int n = 2;

    // Минимальное число
    факультативных задач для студента
    private int size = 0;

    // Вспомогательная переменная
    private String id = "";

    // Строка куда будут записываться id
    задачи, которые выбрал студент
    private String[] real_id;

    // Вспомогательная переменная
    private int j = 0;

    public Raspisanie() {}

    // @Autowired
    public void facultativeRaspisanie(Facultative facultative,Obligatory
obligatory) {
        //-----Выводит на экран список факультативных дисциплин и
их задач-----
        System.out.println("Здравствуйете. Спешим сообщить Вам
новость, что с этого года каждый студент должен выполнять не менее " + n + "
дополнительные занятия по факультативным дисциплинам!\nВот список доступных
факультативов:");
    }
}

```

```

        for (int i = 0; i < facultative.getKolvoObjects();
i++) {
            System.out.println("****" +
facultative.getNameDiscipline()[i]);

            calculateSize(facultative.getKolvoDiscipline()[i]);
            System.out.println("К задачам данного
факультатива относят:");
            while (j < size) {
                System.out.println(facultative.getId()[j]
+ " --- " + facultative.getName()[j]+ " " + facultative.getInfo()[j]);
                j++;
            }
        }
        choice(); // Запускает метод
выбора факультативной задачи
        processChoice(facultative,obligatory);
        // Запускает метод анализа
    }

    // @Autowired
    private void choice() {
        //-----Запрашивает ввод не менее n числа задач из списка
        факультативных задач
        Scanner sc = new Scanner(System.in);
        System.out.println("Введите индекс (-ы) задач которые
вам понравились (в строку через пробел):");
        while ("".equals(id) == true) {
            id = sc.nextLine();
        }
        real_id = id.split("\\s");

        if (real_id.length < n) {
            System.out.println("Вы выбрали меньше " + n + "
факультативов. Попробуйте снова.");
            id = "";
            choice();
        }
    }

    // @Autowired
    private void processChoice(Facultative facultative,Obligatory
obligatory) {
        //-----Анализирует выбранные задачи и запускает метод корректировки
        расписания
        for (int i = 0; i < real_id.length; i++) {
            for (int j = 0; j < facultative.getKolvoId(); j++) {
                if (real_id[i].equals(facultative.getId()[j])) {

                    recalculation(facultative.getKolvoLab()[j],facultative.getKolvoPract()[
j],facultative.getObject()[j],obligatory);
                }
            }
        }
    }
}

```



```

        public void recalculation(int colvo_lab,int colvo_pract,String
name,Obligatory obligatory) {
            for (int i = 0; i < obligatory.getName().length; i++) {
                if (obligatory.getName()[i].equals(name)) {
                    obligatory.setKolvoLab(i,colvo_lab);
                    obligatory.setKolvoPract(i,colvo_pract);
                }
            }
        }

        private void calculateSize(int a) {this.size += a;}

        public void printFacultative(Facultative facultative) {
            for (int i = 0; i < real_id.length; i++) {
                for (int j = 0; j < facultative.getKolvoId(); j++) {
                    if (real_id[i].equals(facultative.getId()[j])) {
                        System.out.println("** " +
facultative.getName()[j]);
                    }
                }
            }
        }
    }
}

```

appAutoscanContext.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"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

    <context:component-scan base-package="springhw.scanbeans"/>
</beans>

```

topicpart.properties

```

obligatory.name=HPP0,Machine Learning
obligatory.lab_topic= 1)Development of loosely coupled program code based on GoF
design patterns, 2)Spring Framework, 3)development of multithreaded applications
using Concurrency, 4)Development of actor-oriented applications in java akka, 1)
Pandas data Analysis, 2) Linear Regression, 3) Decision Trees, 4) Random Forest
method"
obligatory.pract_topic= 1) Basic Design Patterns, 2) Generative Design Patterns, 3)
Structural Design Patterns, 4) Behavioral Design Patterns, 5) Architectural Design

```

Patterns, 1) Quality metrics for classification tasks, 2) Data preprocessing. Feature selection, 3) Error functions in machine learning, 4) Clustering algorithms"

facultative.main.name=Robotics is an applied science that develops automated technical systems, Algorithms and data structures
facultative.name=Machine vision and face recognition in real time, Optimization and retraining by the example of the K-neighbor method, Data analysis and visualization
classification, Working with OpenCV
facultative.info=If you successfully solve this problem you will be credited with one laboratory test on the subject of Machine Learning, If you successfully solve this problem you will be credited with one practical work on the subject of HPP0, If you successfully solve this problem you will be credited with one laboratory test on the subject of HPP0, If you successfully solve this problem you will be credited with one practical work on the subject of Machine Learning"
facultative.id=r486vy,r846cu,a312bg,a764ih
facultative.object=Machine Learning,HPP0,HPP0,Machine Learning
facultative.lab=1,0,1,0
facultative.pract=0,1,0,1

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>nntu.vst.spring</groupId>
    <artifactId>springBase</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <build>
        <plugins>
            <plugin>
                <artifactId>maven-compiler-plugin</artifactId>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
        </plugins>
    </build>

    <dependencies>

        <!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>5.2.4.RELEASE</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.springframework/spring-beans -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-beans</artifactId>
            <version>5.2.4.RELEASE</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
```

```

    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.2.4.RELEASE</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-context-
support -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context-support</artifactId>
      <version>5.2.5.RELEASE</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-aop</artifactId>
      <version>5.2.5.RELEASE</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->
    <dependency>
      <groupId>org.aspectj</groupId>
      <artifactId>aspectjrt</artifactId>
      <version>1.9.5</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->
    <dependency>
      <groupId>org.aspectj</groupId>
      <artifactId>aspectjweaver</artifactId>
      <version>1.9.5</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/net.sf.ehcache/ehcache -->
    <dependency>
      <groupId>net.sf.ehcache</groupId>
      <artifactId>ehcache</artifactId>
      <version>2.10.6</version>
    </dependency>

    <!-- SLF4J/Logback -->
    <dependency>
      <groupId>ch.qos.logback</groupId>
      <artifactId>logback-classic</artifactId>
      <version>1.1.7</version>
    </dependency>

  </dependencies>

</project>

```

Результаты работы программы

```

15:25:12.390 [main] DEBUG
org.springframework.context.support.ClassPathXmlApplicationContext - Refreshing
org.springframework.context.support.ClassPathXmlApplicationContext@4d3167f4

15:25:12.700 [main] DEBUG
org.springframework.context.annotation.ClassPathBeanDefinitionScanner - Identified
candidate component class: file [E:\Play\eclipse
project\www\www\www\www\www\target\classes\springshw\scanbeans\Facultative.class]
15:25:12.704 [main] DEBUG
org.springframework.context.annotation.ClassPathBeanDefinitionScanner - Identified
candidate component class: file [E:\Play\eclipse
project\www\www\www\www\www\target\classes\springshw\scanbeans\Obligatory.class]
15:25:12.705 [main] DEBUG
org.springframework.context.annotation.ClassPathBeanDefinitionScanner - Identified
candidate component class: file [E:\Play\eclipse
project\www\www\www\www\www\target\classes\springshw\scanbeans\Raspisanie.class]
15:25:12.732 [main] DEBUG
org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loaded 7 bean
definitions from class path resource [appAutoscanContext.xml]
15:25:12.756 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean
'org.springframework.context.annotation.internalConfigurationAnnotationProcessor'
15:25:12.793 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean
'org.springframework.context.event.internalEventListenerProcessor'
15:25:12.795 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean
'org.springframework.context.event.internalEventListenerFactory'
15:25:12.796 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean
'org.springframework.context.annotation.internalAutowiredAnnotationProcessor'
15:25:12.802 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean 'fa'
15:25:12.866 [main] DEBUG
org.springframework.core.env.PropertySourcesPropertyResolver - Found key
'facultative.name' in PropertySource 'class path resource [topicpart.properties]'
with value of type String
15:25:12.908 [main] DEBUG
org.springframework.core.env.PropertySourcesPropertyResolver - Found key
'facultative.id' in PropertySource 'class path resource [topicpart.properties]' with
value of type String
15:25:12.909 [main] DEBUG
org.springframework.core.env.PropertySourcesPropertyResolver - Found key
'facultative.main.name' in PropertySource 'class path resource
[topicpart.properties]' with value of type String
15:25:12.910 [main] DEBUG
org.springframework.core.env.PropertySourcesPropertyResolver - Found key
'facultative.info' in PropertySource 'class path resource [topicpart.properties]'
with value of type String
15:25:12.911 [main] DEBUG
org.springframework.core.env.PropertySourcesPropertyResolver - Found key
'facultative.object' in PropertySource 'class path resource [topicpart.properties]'
with value of type String
15:25:12.912 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean 'ob'

```

```

15:25:12.918 [main] DEBUG
org.springframework.core.env.PropertySourcesPropertyResolver - Found key
'obligatory.name' in PropertySource 'class path resource [topicpart.properties]' with
value of type String
15:25:12.918 [main] DEBUG
org.springframework.core.env.PropertySourcesPropertyResolver - Found key
'obligatory.lab_topic' in PropertySource 'class path resource [topicpart.properties]'
with value of type String
15:25:12.919 [main] DEBUG
org.springframework.core.env.PropertySourcesPropertyResolver - Found key
'obligatory.pract_topic' in PropertySource 'class path resource
[topicpart.properties]' with value of type String
15:25:12.920 [main] DEBUG
org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating
shared instance of singleton bean 'ra'

```

Обязательное расписание для Вас:

По предмету НРРО у вас запланировано:

**** 4 лабораторных работ:**

- 1) Development of loosely coupled program code based on GoF design patterns
- 2) Spring Framework
- 3) development of multithreaded applications using Concurrency
- 4) Development of actor-oriented applications in java akka

**** 5 практических работ:**

- 1) Basic Design Patterns
- 2) Generative Design Patterns
- 3) Structural Design Patterns
- 4) Behavioral Design Patterns
- 5) Architectural Design Patterns

По предмету Machine Learning у вас запланировано:

**** 4 лабораторных работ:**

- 1) Pandas data Analysis
- 2) Linear Regression
- 3) Decision Trees
- 4) Random Forest method"

**** 4 практических работ:**

- 1) Quality metrics for classification tasks
- 2) Data preprocessing. Feature selection
- 3) Error functions in machine learning
- 4) Clustering algorithms"

Здравствуйте. Спешим сообщить Вам новость, что с этого года каждый студент должен выполнять не менее 2 дополнительные занятия по факультативным дисциплинам!

Вот список доступных факультативов:

****Robotics is an applied science that develops automated technical systems

К задачам данного факультатива относят:

r486vy --- Machine vision and face recognition in real time If you successfully solve this problem you will be credited with one laboratory test on the subject of Machine Learning

r846cu --- Optimization and retraining by the example of the K-neighbor method If you successfully solve this problem you will be credited with one practical work on the subject of НРРО

****Algorithms and data structures

К задачам данного факультатива относят:

a312bg --- Data analysis and visualization classification If you successfully solve this problem you will be credited with one laboratory test on the subject of НРРО

a764ih --- Working with OpenCV If you successfully solve this problem you will be credited with one practical work on the subject of Machine Learning"

Введите индекс (-ы) задач которые вам понравились (в строку через пробел):

r486vy a312bg

Окончательное расписание для Вас:

По предмету НРРО у вас запланировано:

**** 3 лабораторных работ:**

- 1) Development of loosely coupled program code based on GoF design patterns
- 2) Spring Framework
- 3) development of multithreaded applications using Concurrency

**** 5 практических работ:**

- 1) Basic Design Patterns
- 2) Generative Design Patterns
- 3) Structural Design Patterns
- 4) Behavioral Design Patterns
- 5) Architectural Design Patterns

По предмету Machine Learning у вас запланировано:

**** 3 лабораторных работ:**

- 4) Development of actor-oriented applications in java akka
- 1) Pandas data Analysis
- 2) Linear Regression

**** 4 практических работ:**

- 1) Quality metrics for classification tasks
- 2) Data preprocessing. Feature selection
- 3) Error functions in machine learning
- 4) Clustering algorithms"

**** Machine vision and face recognition in real time**

**** Data analysis and visualization classification**

15:25:31.276 [main] DEBUG

org.springframework.context.support.ClassPathXmlApplicationContext - Closing

org.springframework.context.support.ClassPathXmlApplicationContext@4d3167f4, started

on Sun May 16 15:25:12 MSK 2021

Итоги:

Программа получилась более компактной и удобной для просмотра и внесения изменений. Изменять входные данные стало проще, нет необходимости залезать внутрь программы и править код. Все можно сделать в специальном файле, что позволило программе приблизиться к реальному проекту.