

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
Федерального государственного бюджетного образовательного учреждения высшего  
образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИН \_\_\_\_\_

# ЛАБОРАТОРНАЯ РАБОТА №1

По \_\_\_\_\_ Разработка корпоративных приложений \_\_\_\_\_

Тема \_\_\_\_\_ Обработка форм \_\_\_\_\_

Руководитель

Кульков Я.Ю.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Студент \_\_\_\_\_ ПИН - 121 \_\_\_\_\_  
(группа)

Ермилов М.В.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Муром 2024

## Лабораторная работа №1

Тема: обработка форм.

Цели и задачи: создание Spring-приложения обработки данных с формы.

Ход работы: задание: реализовать Spring-приложение из примера в лабораторной работе. Дополнить приложение и класс Студента следующими данными:

- фамилия;
- группа. На форме вводится только год поступления. Название группы формируется как «ПИН-1» + <последние две цифры года>. Например: год 2022 – группа ПИН-122;
- логин для образовательного портала: формировать по шаблону “student” + <группа> + <id>. Например: student-pin122-12;
- адрес почты в домене института – формировать из фамилии и имени студента и домена @mivlgu.ru Пример: студент Кульков Ярослав – kulkovyaroslav@mivlgu.ru. Красиво оформить страницу с формой и результатом.

### Листинг кода 1 – MainController.java

```
package ru.kafpin.firstttt;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
@Controller
public class MainController {
    @GetMapping("/home")
    public String home(Model model){ return "home"; }

    @GetMapping("/form")
    public String mainForm(Model model){
        model.addAttribute("student", new Student());
        return "main-form";
    }

    @PostMapping("/form")
    public String mainForm(@ModelAttribute Student student, Model
        model){
        model.addAttribute("student", student);
        return "result";
    }
}
```

					МИВлГУ 09.03.04 - 0.007					
Изм.	Лист	№ докум.	Подпись	Дата	Обработка форм			Лит.	Лист	Листов
Разраб.		Ермилов М.В.								
Провер.		Кульков Я.Ю.							2	6
Реценз.								МИ ВлГУ ПИН-121		
Н. Контр.										
Утверд.										

## Листинг кода 2 – Student.java

```
package ru.kafpin.firstttt;
import java.util.HashMap;
import java.util.Map;
public class Student {
    private long id;
    private String name;
    private String surname;
    private Integer yearOfAdmission;
    private String group;
    private String login;
    private String mail;
    public String getMail() {
        return this.toTranslit(this.getSurname()) + this.toTranslit(this.getName()) + "@mivlgu.ru";
    }
    public void setMail(String mail) {
        this.mail = mail;
    }
    public String getGroup() {
        return "ПИИ-1" + takeLast(String.valueOf(this.getYearOfAdmission()),2);
    }
    public void setGroup(String group) {
        this.group = group;
    }
    public String getLogin() {
        return "student-"+this.getGroup()+"-"+(this.getId());
    }
    public void setLogin(String login) {
        this.login = login;
    }
    public String getSurname() {
        return surname;
    }
    public void setSurname(String surname) {
        this.surname = surname;
    }
    public Integer getYearOfAdmission() {
        return yearOfAdmission;
    }
    public void setYearOfAdmission(Integer yearOfAdmission) {
        this.yearOfAdmission = yearOfAdmission;
    }
    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public Student() {
    }
    public String takeLast(String str, int count) {
        if (str == null || str.isEmpty() || count <= 0) {
            return "";
        }
        if (count > str.length()) {
            return str;
        }
        return str.substring(str.length() - count);
    }
    private static final Map<String, String> letters = new HashMap<String, String>();
    static {
        letters.put("А", "A");
        letters.put("Б", "B");
        letters.put("В", "V");
        letters.put("Г", "G");
        letters.put("Д", "D");
        letters.put("Е", "E");
        letters.put("Ё", "E");
        letters.put("Ж", "Zh");
        letters.put("З", "Z");
        letters.put("И", "I");
    }
}
```

					МИВлГУ 09.03.04 – 0.007	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

```

letters.put("Й", "I");
letters.put("К", "K");
letters.put("Л", "L");
letters.put("М", "M");
letters.put("Н", "N");
letters.put("О", "O");
letters.put("П", "P");
letters.put("Р", "R");
letters.put("С", "S");
letters.put("Т", "T");
letters.put("У", "U");
letters.put("Ф", "F");
letters.put("Х", "Kh");
letters.put("Ц", "C");
letters.put("Ч", "Ch");
letters.put("Ш", "Sh");
letters.put("Щ", "Sch");
letters.put("Ъ", "");
letters.put("Ы", "Y");
letters.put("Ь", "");
letters.put("Э", "E");
letters.put("Ю", "Yu");
letters.put("Я", "Ya");
letters.put("а", "a");
letters.put("б", "b");
letters.put("в", "v");
letters.put("г", "g");
letters.put("д", "d");
letters.put("е", "e");
letters.put("ё", "e");
letters.put("ж", "zh");
letters.put("з", "z");
letters.put("и", "i");
letters.put("й", "i");
letters.put("к", "k");
letters.put("л", "l");
letters.put("м", "m");
letters.put("н", "n");
letters.put("о", "o");
letters.put("п", "p");
letters.put("р", "r");
letters.put("с", "s");
letters.put("т", "t");
letters.put("у", "u");
letters.put("ф", "f");
letters.put("х", "h");
letters.put("ц", "c");
letters.put("ч", "ch");
letters.put("ш", "sh");
letters.put("щ", "sch");
letters.put("ъ", "");
letters.put("ы", "y");
letters.put("ь", "");
letters.put("э", "e");
letters.put("ю", "yu");
letters.put("я", "ya");
}
public String toTranslit(String text) {
    StringBuilder sb = new StringBuilder(text.length());
    for (int i = 0; i < text.length(); i++) {
        String l = text.substring(i, i+1);
        if (letters.containsKey(l)) {
            sb.append(letters.get(l));
        }
        else {
            sb.append(l);
        }
    }
    return sb.toString();
}
}

```

### Листинг кода 3 – main-form.html

```

<!DOCTYPE HTML>
<html xmlns:th="https://www.thymeleaf.org">
<head>
    <title>Добавление студента</title>

```

					МИВлГУ 09.03.04 – 0.007	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

```

<link rel="stylesheet" th:href="@{style2.css}" type="text/css">
</head>
<body>
  <div class="page">
    <h1>Информация о студенте</h1>
    <form class="customfrm" action="#" th:action="@{/form}"
    th:object="${student}" method="post">
      <p class="singleRow">Id: <input type="number" th:field="*{id}" /></p>
      <p class="singleRow">Имя студента:
        <input type="text"
        placeholder="Введите имя"
        th:field="*{name}" /></p>
      <p class="singleRow">Фамилия студента:
        <input type="text"
        placeholder="Введите фамилию"
        th:field="*{surname}" /></p>
      <p class="singleRow">Год поступления:
        <input type="number" th:field="*{yearOfAdmission}" />
      </p>
      <p class="singleRow">
        <input type="submit" class="custombtn" value="Добавить" />
        <input type="reset" class="custombtn" value="Очистить поля ввода" />
      </p>
    </form>
    <a href="/home"><button class="custombtn noflexbtn">К домашней странице</button></a>
  </div>
</body>
</html>

```

## Листинг кода 4 – result.html

```

<!DOCTYPE HTML>
<html xmlns:th="https://www.thymeleaf.org">
<head>
  <title>Студент</title>
  <link rel="stylesheet" th:href="@{style2.css}" type="text/css">
</head>
<body>
  <div class="page">
    <h1>Информация о студенте</h1>
    <p th:text="'id: ' + ${student.id}" />
    <p th:text="'Имя: ' + ${student.name}" />
    <p th:text="'Фамилия: ' + ${student.surname}" />
    <p th:text="'Группа: ' + ${student.group}" />
    <p th:text="'Логин: ' + ${student.login}" />
    <p th:text="'Адрес электронной почты: ' + ${student.mail}" />
    <a href="/form"><button class="custombtn noflexbtn">Добавить студента</button></a>
  </div>
</body>
</html>

```

Информация о студенте

Id:

Имя студента:

Фамилия студента:

Год поступления:

Рисунок 1 – Страница добавления информации о студенте

					МИВлГУ 09.03.04 – 0.007	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

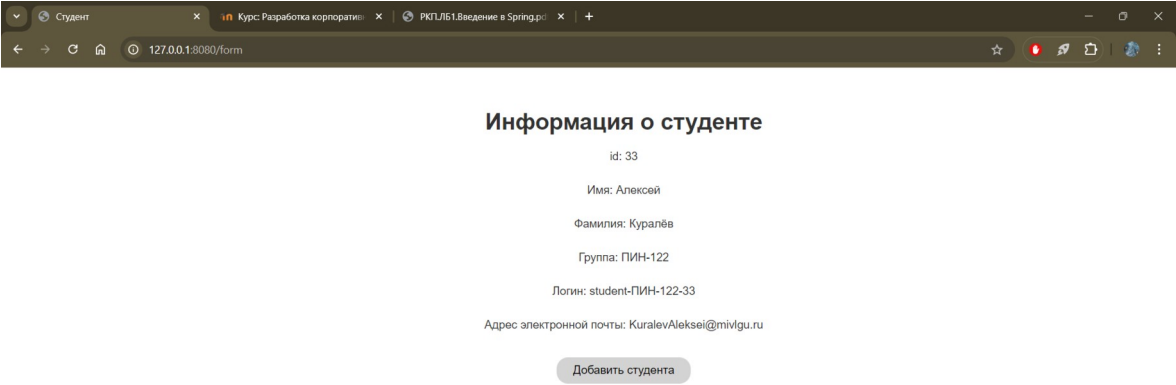


Рисунок 2 – Результат добавления студента

Вывод: в ходе работы создали Spring - приложение обработки данных с формы.