



Ćwiczenie 1:

Należy umożliwić dodawanie nowych studentów poprzez formularz WWW.

Po wypełnieniu formularza, dane mogą być przechowywane w tymczasem strukturze, a lista dodanych studentów wyświetlana jest na ekranie.

Klasa student jest już definiowana.

```
public class Student {  
  
    @NotNull  
    @Length(min = 1, max = 50)  
    private String firstname;  
  
    @NotNull  
    @Length(min = 1, max = 50)  
    private String lastname;  
  
    @NotNull  
    @Email  
    private String emailAddress;  
  
    @NotNull  
    @Past  
    private Date dateOfBirth;  
  
    @NotNull  
    @Past  
    private Date courseStart;  
  
    @NotNull  
    @Future  
    private Date courseFinish;  
  
    @NotNull  
    @Pattern(regexp="^[A-Z]{3}\\-[0-9]{8}\\/[0-9]{2}$")  
    private String idNumber;  
  
    //.. getters and setters  
}
```

Instrukcja rozwiązania

1. Używając wzorca środowiska SpringTool Suite tworzymy nowy projekt Spring MVC
2. Dodajemy klasę Student do projektu.
3. Tworzymy nowy kontroler – StudentController, który dla metody GET wyświetli istniejących studentów, natomiast po wywołaniu metody POST, sprawdzi poprawność przesłanych danych i zapisze je do tymczasowej struktury.
4. Struktura może być zwykłą listą będącą polem kontrolera.



Rozwiązanie

```
@Controller
public class StudentController {

    private List<Student> students = new ArrayList<Student>();

    @RequestMapping(value="/students", method=RequestMethod.GET)
    public String student(Model model) {

        model.addAttribute("students", students);
        model.addAttribute("student", new Student());

        return "students";
    }

    @RequestMapping(value="/students", method=RequestMethod.POST)
    public String addStudent(@Valid Student student, BindingResult results) {

        if (results.hasErrors()) {
            return "students";
        }

        students.add(student);
        return "redirect:/students";
    }
}
```



```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>

<html>
<head>
    <title>Tasks</title>
</head>
<body>
<h1>
    The current list of tasks.
</h1>

<table>
    <c:forEach items="${students}" var="s">
    <tr>
        <td>${s.firstname}</td>
        <td>${s.lastname}</td>
        <td>${s.idNumber}</td>
    </tr>
    </c:forEach>
</table>

<h2>Add new student</h2>

<form:form commandName="student">
    <table>
        <tr>
            <td>First name</td>
            <td>
                <form:input path="firstname" />
                <span><form:errors path="firstname"/></span>
            </td>
        </tr>
        <tr>
            <td>Last name</td>
            <td>
                <form:input path="lastname" />
                <span><form:errors path="lastname"/></span>
            </td>
        </tr>
        <tr>
            <td>Email</td>
            <td>
                <form:input path="emailAddress" />
                <span><form:errors path="emailAddress"/></span>
            </td>
        </tr>
        <tr>
            <td>Date of birth</td>
            <td>
                <form:input path="dateOfBirth" />
                <span><form:errors path="dateOfBirth"/></span>
            </td>
        </tr>
        <tr>
            <td>Course start</td>
            <td>
                <form:input path="courseStart" />
                <span><form:errors path="courseStart"/></span>
            </td>
        </tr>
    </table>
</form:form>
```



```
<tr>
  <td>Course Finish</td>
  <td>
    <form:input path="courseFinish" />
    <span><form:errors path="courseFinish"/></span>

  </td>
</tr>
<tr>
  <td>Student Id</td>
  <td>
    <form:input path="idNumber" />
    <span><form:errors path="idNumber"/></span>

  </td>
</tr>
<tr>
  <td colspan="2">
    <input type="submit" value="Submit" />
  </td>
</tr>
</table>
</form:form>

</body>
</html>
```