Oefening 1: MVC Web Framework

Herschrijf de oefening "Beer Selection Page" in Spring.

Gegeven: **MODEL**
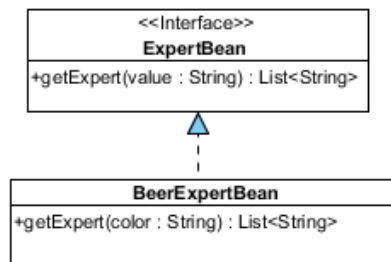

```
package domain;
import java.util.List;
public interface ExpertBean
{
    public List<String> getExpert(String value);
}


package domain;
import java.util.ArrayList;
import java.util.List;
public class BeerExpertBean implements ExpertBean {

  @Override
  public List<String> getExpert(String color) {
     List<String> brands = new ArrayList<>();
     if (color == null)
        return brands;

     switch (color.toLowerCase().trim()) {
        case "light":
           brands.add("Hoegaarden");
           brands.add("Brugs Witbier");
           break;
        case "brown":
           brands.add("Westmalle bruin");
           brands.add("Leffe bruin");
           break;
        case "dark":
           brands.add("Affligem Donker");
           break;
     }
     return brands;
  }
}
```

```
              ColorBean
<<Property>> -colorsList : List<String>
+ColorBean()
```

package domain;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class ColorBean {
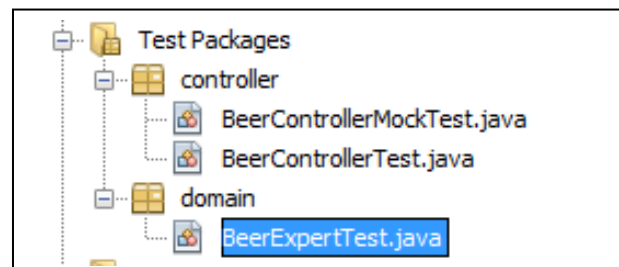
    private final List<String> colorsList;

    public ColorBean() {
        colorsList = new ArrayList<>(Arrays.asList(new String[]{"light","brown","dark"}));
    }

    public List<String> getColorsList() {
        return colorsList;
    }
}

JUNIT
package domain;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;
import java.util.List;
import junit.framework.Assert;
import org.junit.Before;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;


```
Test Packages
    controller
        BeerControllerMockTest.java
        BeerControllerTest.java
    domain
        BeerExpertTest.java
```

@RunWith(value = Parameterized.class)
public class **BeerExpertTest** {

 private String input;
 private List<String> expResult;
 private ExpertBean beerExpert;

 private static List<String>
        resultLight = new ArrayList(Arrays.asList(new String[]{"Hoegaarden", "Brugs Witbier"})),
        resultBrown = new ArrayList<>(Arrays.asList(new String[]{"Westmalle bruin", "Leffe bruin"})),
        resultDark = new ArrayList<>(Arrays.asList(new String[]{"Affligem Donker"})),
        resultEmpty = new ArrayList<>();

  @Parameters
  public static Collection<Object[]> getParameters() {
    return Arrays.asList(new Object[][]{
        {"light", resultLight},{"brown", resultBrown},
        {"dark", resultDark},{"LIGHT", resultLight},
        {"Brown", resultBrown},{"DaRk", resultDark},
        {" light   ", resultLight}, {"l ight", resultEmpty},
        {null, resultEmpty},{"", resultEmpty},
        {"   ", resultEmpty}, {"abc", resultEmpty}
    });
  }

```
  public BeerExpertTest(String input, List<String> expResult) {
    this.input = input;
    this.expResult = expResult;
  }

  @Before
  public void before()
  {
    beerExpert = new BeerExpertBean();
  }

  @Test
  public void colorBeerTest()
  {
    Assert.assertEquals(expResult, beerExpert.getExpert(input));
  }
}
```

Gevraagd:

WebConfig
SpringMvcInitializer

CONTROLLER:
       BeerController
       BeerCommand

VIEW:
       formView.jsp
       resultView.jsp

testklasse:
       BeerControllerTest
       BeerControllerMockTest

Project Properties - Spring4_Web_beerOef_Config

Categories:
- Sources
- Frameworks
- Spring Framework
- Libraries
- JavaScript Libraries
  - npm
  - Bower
  - CDNJS
- CSS Preprocessors
- Build
  - Compiling
  - Packaging
  - Documenting
- Run
- License Headers
- Formatting

Server: GlassFish Server 4.1.1

Java EE Version: Java EE 7 Web

Context Path: /Spring4_Web_beerOef_Config

☑ Display Browser on Run

Relative URL: beer.htm **beer.htm**

Specify the URL relative to the context path to run, eg. /admin/login.jsp

Browser: IDE's default browser

☑ Deploy on Save

If selected, files are compiled and deployed when you save them.
This option saves you time when you run or debug your application in the IDE.

VM Options:

(used for running main classes or unit tests; e.g. -Xms10m)

OK   Cancel   Help

①

Web browser → beer.htm get → DispatcherServlet ② ← → **BeerController** Controller

⑥

forward ModelAndView

⑤ rendered HTML

④ **ModelAndView**
(model =

| beerCommand | object van BeerCommand |
| colorsList | List<String> |

③

)
(view = formView)

/WEB-INF/jsp/formView.jsp

View

ColorBean

← → C localhost:8080/Spring4_Web_beerOef_Config/beer.htm

Chose color: light ▼ OK

Diagram: post 1 /beer.htm — Web browser → DispatcherServlet → (2) → BeerController → BeerCommand (Controller); BeerController → (3) Get Model → ExpertBean (service/business objects) — Model; ModelAndView (view = resultView) (4); (5) forward model / rendered HTML → /WEB-INF/jsp/resultView.jsp (View); (6)

localhost:8080/Spring4_Web_beerOef_Config/beer.htm

Got beer color light
try:
Hoegaarden
Brugs Witbier

controller.BeerControllerTest
100,00 %
Both tests passed. (0,939 s)

controller.BeerControllerMockTest
100,00 %
The test passed. (0,934 s)