

КАФЕДРА №

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

---

должность, уч. степень, звание

---

подпись, дата

---

инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

Разработка AJAX приложения с использованием Hibernate

по дисциплине: Технология разработки серверных информационных систем

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

---

подпись, дата

---

инициалы, фамилия

Санкт-Петербург  
2023

**Текст и вариант задания:**

13. Торговля акциями на бирже.

**Описание разрабатываемого продукта:**

В программе создан сервис для покупки, продажи и просмотра портфеля акций.

**Задание на лабораторную работу.**

- 1 Добавьте базу данных, модель данных, JPA репозитории и сервисы.
- 2 Реализуйте операции чтения/записи из одной из предыдущих лабораторных работ (3 или 5) с помощью JPA

**Текст основных фрагментов кода:**

Itemobj.java

```
package com.example.ThymeleafProject;
```

```
import jakarta.persistence.Entity;
```

```
import jakarta.persistence.Id;
```

```
@Entity
```

```
public class ItemObj {
```

```
    @Id
```

```
    private int stockID;
```

```
    private String stock_name;
```

```
    private String purchase_date;
```

```
    public ItemObj() {}
```

```
    public ItemObj(int stockID, String stock_name, String purchase_date) {
```

```
        this.stockID = stockID;
```

```
        this.stock_name = stock_name;
```

```
        this.purchase_date = purchase_date;
    }
}
```

```
public int getStockID() {
    return stockID;
}
```

```
public void setStockID(int stockID) {
    this.stockID = stockID;
}
```

```
public String getStock_name() {
    return stock_name;
}
```

```
public void setStock_name(String stock_name) {
    this.stock_name = stock_name;
}
```

```
public String getPurchase_date() {
    return purchase_date;
}
```

```
public void setPurchase_date(String purchase_date) {
    this.purchase_date = purchase_date;
}
```

```
@Override
```

```
public String toString() {
```

```

        return "ItemObj{" +
            "stockID=" + stockID +
            ", stock_name='" + stock_name + "\" +
            ", purchase_date='" + purchase_date + "\" +
            '}'';
    }
}

```

Itemobjrepository

```
package com.example.ThymeleafProject;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface ItemObjRepository extends JpaRepository<ItemObj, Integer> {
}

```

Logincontroller

```
package com.example.ThymeleafProject;
```

```
import jakarta.servlet.ServletException;
```

```
import jakarta.servlet.http.HttpServletRequest;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
@Controller
```

```
public class LoginController {

```

```
@GetMapping("/login")
```

```
public String login() {
```

```
    return "login";
```

```
}
```

```
@PostMapping("/logout")
```

```
public String logout(HttpServletRequest request) throws ServletException {
```

```
    request.logout();
```

```
    return "redirect:/login";
```

```
}
```

```
}
```

Stockcontroller

```
package com.example.ThymeleafProject;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.*;
```

```
@Controller
```

```
@RequestMapping("/stocks")
```

```
public class StockController {
```

```
    private final ItemObjRepository itemObjRepository;
```

```
@Autowired
```

```
public StockController(ItemObjRepository itemObjRepository) {
```

```
        this.itemObjRepository = itemObjRepository;
    }
}
```

@GetMapping

```
public String getAllStocks(Model model) {
    model.addAttribute("stocks", itemObjRepository.findAll());
    return "stocks";
}
```

@PostMapping

```
public String addStock(@ModelAttribute ItemObj stock) {
    itemObjRepository.save(stock);
    return "redirect:/stocks";
}
```

@GetMapping("/{id}")

```
public String getStock(@PathVariable Integer id, Model model) {
    itemObjRepository.findById(id)
        .ifPresent(stock -> model.addAttribute("stock", stock));
    return itemObjRepository.findById(id).isPresent() ? "stock" : "redirect:/stocks";
}
```

@PostMapping("/delete/{id}")

```
public String deleteStock(@PathVariable Integer id) {
    itemObjRepository.deleteById(id);
    return "redirect:/stocks";
}
```

```
}
```

ThymeleafProjectApplication

```
package com.example.ThymeleafProject;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
import org.springframework.context.MessageSource;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.springframework.context.support.ReloadableResourceBundleMessageSource;
```

```
import org.springframework.web.servlet.LocaleResolver;
```

```
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
```

```
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
```

```
import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;
```

```
import org.springframework.web.servlet.i18n.SessionLocaleResolver;
```

```
import java.util.Locale;
```

```
import org.apache.catalina.Context;
```

```
import org.apache.tomcat.util.descriptor.web.FilterDef;
```

```
import org.apache.tomcat.util.descriptor.web.FilterMap;
```

```
import org.springframework.boot.web.embedded.tomcat.TomcatServletWebServerFactory;
```

```
import org.springframework.boot.web.servlet.server.ServletWebServerFactory;
```

```
import org.springframework.web.filter.CharacterEncodingFilter;
```

@SpringBootApplication

```
public class ThymeleafProjectApplication implements WebMvcConfigurer {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(ThymeleafProjectApplication.class, args);
```

```
    }
```

```

// Бин для MessageSource

@Bean

public MessageSource messageSource() {

    ReloadableResourceBundleMessageSource messageSource = new
ReloadableResourceBundleMessageSource();

    messageSource.setBasename("classpath:messages");

    messageSource.setDefaultEncoding("UTF-8");

    return messageSource;

}


// Бин для LocaleResolver

@Bean

public LocaleResolver localeResolver() {

    SessionLocaleResolver slr = new SessionLocaleResolver();

    slr.setDefaultLocale(Locale.ENGLISH); // Английский язык по умолчанию

    return slr;

}


// Бин для LocaleChangeInterceptor

@Bean

public LocaleChangeInterceptor localeChangeInterceptor() {

    LocaleChangeInterceptor lci = new LocaleChangeInterceptor();

    lci.setParamName("lang");

    return lci;

}


// Добавление интерцептора для перехвата изменений локали

@Override

public void addInterceptors(InterceptorRegistry registry) {

```



```

        registry.addInterceptor(localeChangeInterceptor());
    }

    // Настройка кодировки UTF-8 для Tomcat

    @Bean

    public ServletWebServerFactory servletContainer() {

        TomcatServletWebServerFactory tomcat = new
TomcatServletWebServerFactory() {

            @Override

            protected void postProcessContext(Context context) {

                FilterDef filterDef = new FilterDef();

                filterDef.setFilterName("setCharacterEncodingFilter");

filterDef.setFilterClass(CharacterEncodingFilter.class.getName());

                filterDef.addInitParameter("encoding", "UTF-8");

                filterDef.addInitParameter("forceEncoding", "true");

                context.addFilterDef(filterDef);


                FilterMap filterMap = new FilterMap();

                filterMap.setFilterName("setCharacterEncodingFilter");

                filterMap.addURLPattern("/*");

                context.addFilterMap(filterMap);

            }

        };

        return tomcat;
    }
}

```

Websecurityconfig

```

package com.example.ThymeleafProject;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.http.HttpMethod;

import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManag
erBuilder;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import
org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity

public class WebSecurityConfig {

    // Определите цепочку фильтров безопасности

    @Bean

    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

        http

            .authorizeRequests(authorize -> authorize

                .requestMatchers("/", "/home", "/login", "/style.css", "/js/**",
"/images/**").permitAll()

                .requestMatchers(HttpMethod.POST, "/stocks").authenticated()

                .anyRequest().authenticated()

            )

```

```

        .formLogin(form -> form
            .loginPage("/login")
            .defaultSuccessUrl("/stocks", true)
            .permitAll()
        )
        .logout(logout -> logout.permitAll());
    return http.build();
}

```

// Настройка пользователей в памяти

@Bean

```

public InMemoryUserDetailsManager userDetailsService() {
    UserDetails user = User.withDefaultPasswordEncoder()
        .username("user")
        .password("password")
        .roles("ADMIN")
        .build();
    return new InMemoryUserDetailsManager(user);
}

```

// Настройка менеджера аутентификации

@Autowired

```

public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
    auth
        .inMemoryAuthentication()
        .withUser("user")
        .password("{noop}password")
        .roles("ADMIN");
}

```

```
}
```

Style.css

```
/* ОСНОВНЫЕ СТИЛИ */
```

```
body {  
    font-family: 'Arial', sans-serif;  
    margin: 0;  
    padding: 0;  
    background-color: #f4f4f4;  
    color: #333;  
}
```

```
.container {  
    width: 90%;  
    max-width: 1200px;  
    margin: 20px auto;  
    padding: 15px;  
    background: white;  
    border-radius: 10px;  
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
    overflow: hidden;  
}
```

```
h1, h2 {  
    color: #5cb85c; /* Зеленый цвет */  
    text-align: center;  
    margin-bottom: 20px;  
}
```

```
/* Стили таблиц */
```

```
table {  
    width: 100%;  
    border-collapse: collapse;  
    margin-top: 20px;  
}
```

```
th, td {  
    padding: 10px;  
    border: 1px solid #ddd;  
    text-align: left;  
}
```

```
th {  
    background-color: #5cb85c; /* Зеленый цвет */  
    color: white;  
}
```

```
tr:nth-child(even) {  
    background-color: #f2f2f2;  
}
```

```
tr:hover {  
    background-color: #e2e2e2;  
}
```

```
/* Стили форм и элементов ввода */
```

```
form {
```

```
margin-top: 20px;
text-align: left;
padding: 20px;
background: #fff;
border-radius: 5px;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

label {
    margin-top: 10px;
    display: block;
    font-weight: bold;
}

input[type="text"],
input[type="number"],
input[type="date"],
input[type="password"],
select {
    width: 100%;
    padding: 8px;
    margin-top: 5px;
    border: 1px solid #ddd;
    border-radius: 4px;
    box-sizing: border-box;
}

/* Стили кнопок */
.delete-button, .details-button, button, input[type="submit"] {
```

```
padding: 10px 15px;
border: none;
border-radius: 4px;
cursor: pointer;
margin-top: 10px;
width: 100%;
text-align: center;
}

.details-button { /* Зеленая кнопка для Details */
    background-color: #5cb85c;
    color: white;
}

.details-button:hover {
    background-color: #4cae4c;
}

.delete-button { /* Красная кнопка для Delete */
    background-color: #d9534f;
    color: white;
}

.delete-button:hover {
    background-color: #c9302c;
}

.submit-button { /* Зеленая кнопка для Submit */
    background-color: #5cb85c;
```

```
        color: white;
    }

.submit-button:hover {
    background-color: #4cae4c;
}

/* Стили для страницы входа и информации об акции */
.login-form {
    max-width: 400px;
    margin: 50px auto;
    padding: 20px;
    background: white;
    border-radius: 10px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    text-align: center;
}

.login-form input[type="text"],
.login-form input[type="password"] {
    margin-bottom: 15px;
}

.login-form label {
    margin-bottom: 5px;
    display: block;
    text-align: left;
}

.stock-info {
```



```
max-width: 400px;
margin: 0 auto;
padding: 20px;
background: white;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}
```

```
/* Стили ссылок */
```

```
a {
    color: #4a90e2;
    text-decoration: none;
}
```

```
a:hover {
    text-decoration: underline;
}
```

```
/* Медиа-запросы для адаптивного дизайна */
```

```
@media (max-width: 768px) {
    .container, .login-form, .stock-info {
        width: 95%;
        padding: 10px;
    }
}
```

Login.html

```
<!DOCTYPE html>
```

```

<html xmlns:th="http://www.thymeleaf.org">

<head>

    <meta charset="UTF-8">

    <title th:text="#{login.title}">Вход в систему управления акциями</title>

    <link rel="stylesheet" type="text/css" th:href="@{/style.css}" />

</head>

<body>

<div class="login-form">

    <h2 th:text="#{login.header}">Вход в систему управления акциями</h2>

    <form action="/login" method="post" th:action="@{/login}">

        <input type="hidden" th:name="${_csrf.parameterName}" th:value="${_csrf.token}" />

        <div>

            <label th:text="#{login.username}">Имя пользователя:</label>

            <input type="text" name="username" required>

        </div>

        <div>

            <label th:text="#{login.password}">Пароль:</label>

            <input type="password" name="password" required>

        </div>

        <div>

            <input type="submit" class="submit-button" th:value="#{login.submit}"
value="Войти">

        </div>

    </form>

</div>

</body>

</html>

```

Stock.html

```

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

  <meta charset="UTF-8">

  <title th:text="#{stock.title}">Акция</title>

  <link rel="stylesheet" type="text/css" th:href="@{/style.css}" />

</head>

<body>

<div class="container">

  <h1 th:text="#{stock.info}">Информация об акции</h1>

  <p><strong th:text="#{stock.name}">Название акции:</strong> <span
th:text="${stock.stock_name}"></span></p>

  <p><strong th:text="#{stock.id}">ID акции:</strong> <span
th:text="${stock.stockID}"></span></p>

  <p><strong th:text="#{stock.date}">Дата покупки:</strong> <span
th:text="${stock.purchase_date}"></span></p>

  <a th:href="@{/stocks}" th:text="#{stock.return}">Вернуться к списку акций</a>

</div>

</body>

</html>

```

## Stocks.html

```

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

  <meta charset="UTF-8">

  <title th:text="#{stocks.title}">Акции</title>

  <link rel="stylesheet" type="text/css" th:href="@{/style.css}" />

</head>

<body>

```

```

<div class="container">

  <h1 th:text="#{stocks.header}">Акции</h1>

  <table>

    <tr>

      <th th:text="#{stocks.name}">Название акции</th>

      <th th:text="#{stocks.id}">ID акции</th>

      <th th:text="#{stocks.date}">Дата покупки</th>

      <th></th>

    </tr>

    <tr th:each="stock : ${stocks}">

      <td th:text="${stock.stock_name}"></td>

      <td th:text="${stock.stockID}"></td>

      <td th:text="${stock.purchase_date}"></td>

      <td>

        <a th:href="@{/stocks/{id}(id=${stock.stockID})}" class="submit-button"
th:text="#{stocks.details}">Подробнее</a>

        <form th:action="@{/stocks/delete/{id}(id=${stock.stockID})}" method="post">

          <input type="submit" class="delete-button" th:value="#{stocks.delete}"
value="Удалить">

        </form>

      </td>

    </tr>

  </table>

  <h2 th:text="#{stocks.add}">Добавить акцию</h2>

  <form th:action="@{/stocks}" method="post">

    <input type="text" name="stock_name" th:placeholder="#{stocks.stockName}" required>

    <input type="text" name="stockID" th:placeholder="#{stocks.stockId}" required>

    <input type="date" name="purchase_date" th:placeholder="#{stocks.purchaseDate}"
required>

    <input type="submit" class="submit-button" th:value="#{stocks.submit}"
value="Добавить">

```

```
</form>
</div>
</body>
</html>
```

```
messages_en.properties
login.title=Stock Management System Login
login.header=Login to Stock Management System
login.username=Username
login.password=Password
login.submit=Login
stock.title=Stock
stock.info=Stock Information
stocks.header=Stocks
stocks.name=Name
stocks.id=ID
stocks.date=Purchase Date
stocks.details=Details
stocks.delete=Delete
stocks.add=Add New Stock
stocks.stockName=Stock Name
stocks.stockId=Stock ID
stocks.purchaseDate=Purchase Date
stocks.submit=Submit
stock.name=Name
stock.id=ID
stock.date=Purchase date
```

stock.return=Return to stock list

stocks.title=Stock

messages\_ru.properties

login.title=Д' Ñ..Д³⁄₄Д' Д² Ñ • Д,Ñ • Ñ, ДμД¹⁄₄Ñf Ñf ДᵢÑ€ Д°Д²Д»ДμД¹⁄₂Д,Ñ • Д°Д°Ñ† Д,Ñ • Д¹⁄₄Д,

login.header=Д' Ñ..Д³⁄₄Д' Д² Ñ • Д,Ñ • Ñ, ДμД¹⁄₄Ñf Ñf ДᵢÑ€ Д°Д²Д»ДμД¹⁄₂Д,Ñ • Д°Д°Ñ† Д,Ñ • Д¹⁄₄Д,

login.username=Д~ Д¹⁄₄Ñ • ДᵢД³⁄₄Д»Ñ(Д·Д³⁄₄Д²Д°Ñ, ДμД»Ñ •

login.password=ДŸД°Ñ€ Д³⁄₄Д»Ñ(

login.submit=Д' Д³⁄₄Д¹Ñ, Д,

stock.title=Д • Д°Ñ† Д,Ñ •

stock.info=Д~ Д¹⁄₂Ñ,, Д³⁄₄Ñ€ Д¹⁄₄Д°Ñ† Д,Ñ • Д³⁄₄Д± Д°Д°Ñ† Д,Д,

stocks.header=Д • Д°Ñ† Д,Д,

stocks.name=Д • Д°Д·Д²Д°Д¹⁄₂Д,Дμ

stocks.id=Д~ Д ДμД¹⁄₂Ñ, Д,Ñ,, Д,Д°Д°Ñ, Д³⁄₄Ñ€

stocks.date=Д'' Д°Ñ, Д° ДᵢД³⁄₄Д°Ñf ДᵢД°Д,

stocks.details=ДŸД³⁄₄Д'Ñ€ Д³⁄₄Д±Д¹⁄₂ДμДμ

stocks.delete=Д£Д Д°Д»Д,Ñ, Ñ(

stocks.add=Д'' Д³⁄₄Д±Д°Д²Д,Ñ, Ñ(Д°Д°Ñ† Д,ÑŹ

stocks.stockName=Д • Д°Д·Д²Д°Д¹⁄₂Д,Дμ Д°Д°Ñ† Д,Д,

stocks.stockId=ID Д°Д°Ñ† Д,Д,

stocks.purchaseDate=Д'' Д°Ñ, Д° ДᵢД³⁄₄Д°Ñf ДᵢД°Д,

stocks.submit=Дž Ñ, ДᵢÑ€ Д°Д²Д,Ñ, Ñ(

stock.name=Д • Д°Д·Д²Д°Д¹⁄₂Д,Дμ Д°Д°Ñ† Д,Д,

stock.id=ID Д°Д°Ñ† Д,Д,

stock.date=Д'' Д°Ñ, Д° ДᵢД³⁄₄Д°Ñf ДᵢД°Д,

stock.return=Д' ДμÑ€ Д¹⁄₂Ñf Ñ, Ñ(Ñ • Ñ • Д°Ñ • ДᵢД,Ñ • Д°Ñf Д°Д°Ñ† Д,Д¹

stocks.title=Д • Д°Ñ† Д,Д,