

**Бугайов Олександр Сергійович**

*студент*

*Національного технічного університету України*

*«Київський політехнічний інститут імені Ігоря Сікорського»*

**Бугаев Александр Сергеевич**

*студент*

*Национального технического университета Украины*

*«Киевский политехнический институт имени Игоря Сикорского»*

**Buhaiov Oleksandr**

*student of the*

*National technical university of Ukraine*

*«Igor Sikorsky Kyiv Polytechnic Institute»*

## МОЖЛИВОСТІ ФРЕЙМВОРКУ SPRING ДЛЯ ПОБУДОВИ КОРПОРАТИВНИХ СИСТЕМ ТА ЙОГО ІНТЕГРАЦІЯ З ІНШИМИ ФРЕЙМВОРКАМИ

## ВОЗМОЖНОСТИ ФРЕЙМВОРКА SPRING ДЛЯ ПОСТРОЕНИЯ КОРПОРАТИВНЫХ СИСТЕМ И ЕГО ИНТЕГРАЦИЯ ДРУГИМИ ФРЕЙМВОРКАМИ

## SPRING FRAMEWORK CAPABILITIES OF BUILDING ENTERPRISE SYSTEMS AND ITS INTEGRATION WITH OTHER FRAMEWORKS

**Анотація.** Досліджені архітектура та можливості Java-фреймворку Spring в контексті побудови складних корпоративних систем, розглянуто можливості інтеграції з іншими популярними фреймворками.

**Ключові слова:** корпоративні системи, Java, Spring, фреймворк.

**Аннотация.** Исследованы архитектура и возможности Java-фреймворка Spring в контексте построения сложных корпоративных систем, рассмотрены возможности интеграции с другими популярными фреймворками.

**Ключевые слова:** корпоративные системы, Java, Spring, фреймворк.

**Summary.** Architecture and capabilities of Java Spring framework were investigated among with integration possibility with other frameworks.

**Key words:** enterprise systems, Java, Spring, framework.

**В**ведення. З розвитком мережевих технологій та відповідним швидким розвитком електронної комерції (e-commerce) онлайн-покупки стають для клієнтів одним з головних напрямків. Як наслідок, все більше підприємств впроваджують різноманітні онлайн-магазини для реалізації своєї продукції. В основі багатьох таких систем лежить клієнт-серверна архітектура, що поділяє систему на два головних компонента: серверну

частину та клієнтську. Серверна частина відповідає за обробку, збереження та зміну певних даних, що зберігаються системою, відповідно до запитів, що надсилаються клієнтською частиною. У мережевій моделі клієнтом частіше за все виступає веб-сторінка, а взаємодія з сервером відбувається засобами HTTP(S) протоколу.

З розвитком технологій розробки з'являються нові засоби для вдосконалення та спрощення про-

цесу створення програмних систем. Фреймворки є одним з видів таких засобів. Їх можна розглядати як набір класів та/або функцій, що допомагають більш ефективно розробляти застосунки, не розробляючи кожного разу одні й ті самі компоненти. Spring – фреймворк, що полегшує розробку Java застосунків. При поєднанні зі стандартами Java EE, Spring дозволяє будувати багаторівневі системи, що є більш гнучкими для модифікації та розширення.

### Фреймворк Spring

**Архітектура.** Spring надає практично усі необхідні для розробки засоби на всіх рівнях застосунку. Він є модульним, таким чином немає необхідності включати в проект зайві компоненти, які не потрібні при розробці. Усі можливості Spring організовані у 20 модулів. Архітектура фреймворку зображена на рис. 1 [1].

**Spring Core:** це основна частина Spring яка складається з наступних модулів – Core, Beans, Context і Expression Language. Короткий опис виглядає наступним чином:

- Core: Це фундаментальний модуль структури з IOC та ін'єкцією залежностей з паттерном проектування одинак.
- Bean: Цей модуль – реалізація фабричного паттерну через BeanFactory. BeanFactory застосовує IOC, щоб відокремити конфігурацію програми та специфікацію залежностей від фактичної логіки програми.
- Context: він (ApplicationContext) розширює концепцію BeanFactory, додаючи підтримку для інтер-

націоналізації, подій життєвого циклу застосунка та валідацію. Також включає в себе такі сервіси як E-mail, JNDI-доступ, EJB-інтеграція, Remoting та Scheduling.

- Expression Language: Spring3.0 представляє нову мову вираження – Spring ExpressionLanguage (SpEL) Це потужна мова вираження на основі Java Server Pages (JSP) Expression Language(EL). Вона використовується для написання запитів до bean'ів для доступу та управління їх властивостями.

**Data Access:** Це основна частина рівня доступу до бази даних і складається з наступних модулів – JDBC, ORM, OXM, JMS та модуль управління транзакціями:

- JDBC: модулі JDBC забезпечують абстракцію від безпосереднього використання JDBC, що дозволяє зменшити складність його традиційного використання.
- ORM: модуль ORM забезпечує узгодженість/переносимість коду незалежно від обраної технології доступу до даних, заснованих на концепції об'єктно-орієнтованого відображення, таких як Hibernate та інші.
- OXM: OXM представляє Spring3.0 як окремий модуль. Він використовується для перетворення об'єкта в XML формат і навпаки. Spring OXM надає уніфікований API для доступу до будь-якого з OXM (Castor, XStream, JiBX, Java API для XML і XmlBeans) фреймворків.
- JMS: модуль JMS забезпечує зменшення кількості рядків коду для надсилання та отримання повідомлень. API піклується про роботу JMS та обробку виключень.

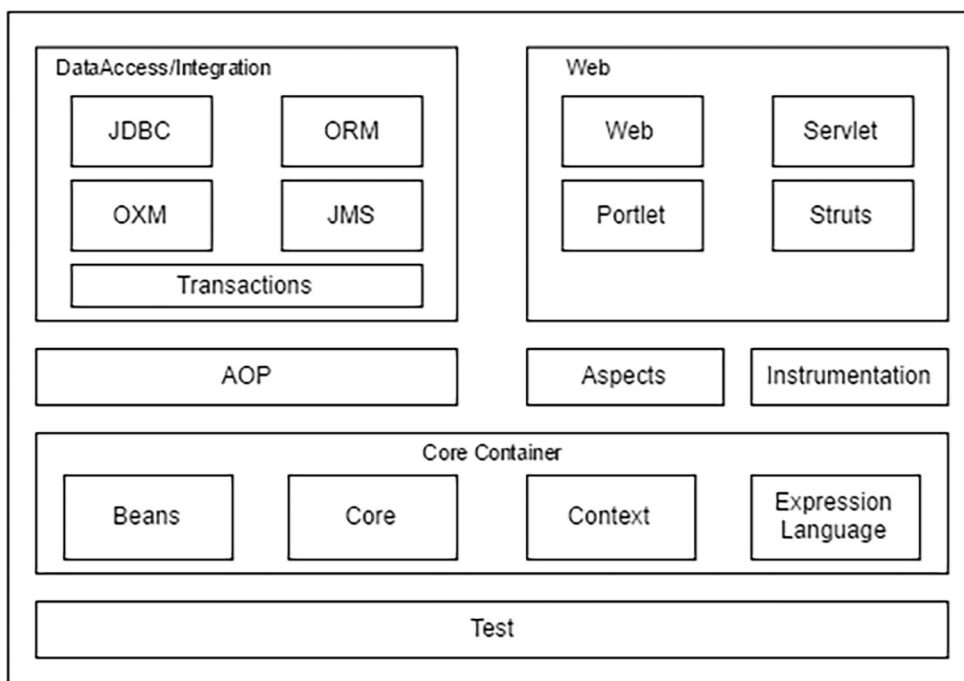


Рис. 1. Архітектура фреймворку Spring

• Transaction: модуль транзакцій підтримує програмне та декларативне керування транзакціями для POJO класів.

**Web:** Це основна частина веб-рівня і складається з наступних модулів – Web, Web-Servlet, Web-Struts і WebPortlet.

• Web: Цей модуль надає основні базові веб-орієнтовані функції інтеграції, такі як багато-частинне завантаження файлів та ініціалізація контейнера IoC за допомогою сервлет-слухачів та веб-орієнтованого контексту застосунка.

• Web-Servlet: Модуль Web-Servlet містить MVC-реалізацію для веб-застосунків. Він забезпечує всі інші функції MVC, включаючи теги інтерфейсу користувача та перевірку даних.

• Web-Struts: модуль Web-Struts містить допоміжні класи для інтеграції класичного веб-рівня Struts в застосунок на Spring. Містить класи для інтеграції Struts1.x та Struts2.

• Web-Portlet: Модуль Web-Portlet забезпечує реалізацію MVC для використання в портлетному середовищі та є аналогічним за функціональністю до модуля Web-Servlet.

Іншими ключовими компонентами є:

• AOP: містить API для аспектно-орієнтованих програмних реалізацій на різних рівнях. Надається можливість ввести новий функціонал в існуючий код, не змінюючи його.

• Aspect: Окремий модуль Aspects забезпечує інтеграцію з AspectJ.

• Test: модуль інструментальних засобів надає підтримку класним інструментам та реалізаціям класу навантажувачів для використання на деяких серверах додатків.

Spring – безкоштовний фреймворк з відкритим програмним кодом, що пропонує програмістам багато функцій. Одними з найголовніших особливостей є Inversion of Control (IoC), аспектно-орієнтоване програмування та Spring MVC, що являє собою окремий фреймворк, який може бути використаний разом з іншими фреймворками. За допомогою аспектно-орієнтованого програмування різноманітні проблеми можуть бути легко розділеними. У Spring аспекти поєднуються разом за допомогою файлу spring.xml, а їх кодування є добре розділеним.

**Enterprise-система з використанням Spring** та інших фреймворків. Spring Framework при побудові enterprise-застосунків може використовуватись разом з багатьма іншими фреймворками такими як Struts та Hibernate.

**Фреймворк Struts.** Даний фреймворк розділяє систему на три рівня: рівень представлення, моделі та контролера, що відповідають архітектурному

патерну MVC. Модель складається з компонентів JavaBeans чи EJB; представлення реалізується у вигляді JSP-файлів; контролер реалізується компонентом ActionServlet. Типова архітектура застосунку з використанням Struts зображена на рис. 2 [2].

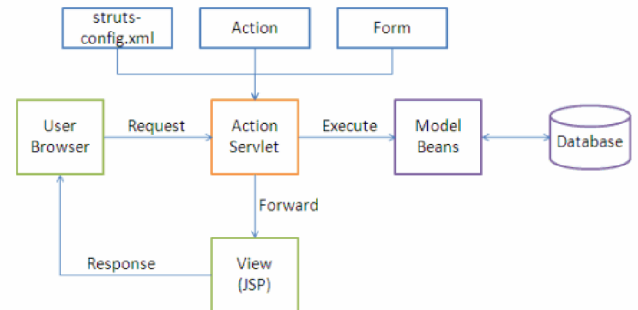


Рис. 2. Типова архітектура Struts

**Фреймворк Hibernate.** Даний фреймворк зменшує складність та трудомісткість маніпуляцій з даними в SQL базах даних. Він ефективно відображає класи Java на таблиці у БД.

**Інтеграція Struts, Spring та Hibernate.** Struts, Spring та Hibernate можуть бути об'єднані для ефективного застосування в enterprise-застосунках (рис. 4). Структура Struts з ефективною MVC може допомогти у розробці рівня презентації. Фреймворк Spring може спростити побудову бізнес-логіки. Фреймворк Hibernate може допомогти у роботі з даними, що знаходяться в системі, а також у збереженні та отриманні даних, що знаходяться в БД.

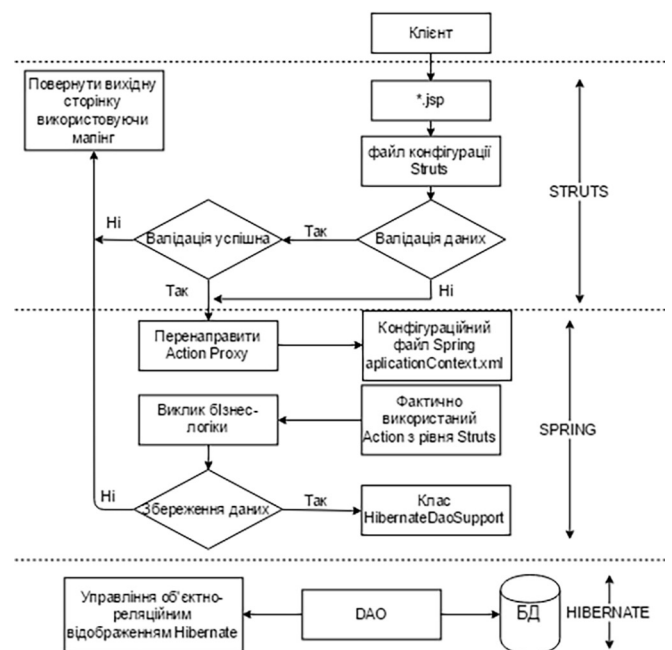


Рис. 3. Принцип взаємодії Hibernate, Struts, Spring

**Висновки.** Spring — потужний фреймворк для побудови enterprise-застосунків. Завдяки добре продуманій архітектурі його можна використовувати разом з багатьма іншими фреймворками, такими як Struts чи Hibernate, що допомагає зменшити зв'язність та більш чітко відокремити компоненти один від одного. Також слід відмітити легкість даних компонентів, що спрощує їх використання. Дозволяється використову-

вати прості сервери застосунків, наприклад, Tomcat. Враховуючи описаний сценарій, в якому присутня структура struts2, що може бути використана в якості веб-представлення, Spring можна ефективно використовувати для побудови корпоративної програми. Фреймворк Spring може бути легко інтегрованим з будь-яким ORM-інструментом, таким як Hibernate, за допомогою XML.

#### Література

1. Spring Framework — Architecture [Електронний ресурс] / TutorialsPoint — Режим доступу до ресурсу: [https://www.tutorialspoint.com/spring/spring\\_architecture.htm](https://www.tutorialspoint.com/spring/spring_architecture.htm).
2. Muthuraman M. Struts MVC Architecture Tutorial [Електронний ресурс] / Meeyappan Muthuraman / DZone. — 2012. — Режим доступу до ресурсу: <https://dzone.com/tutorials/java/struts/struts-tutorial/struts-mvc-architecture-tutorial.html>.
3. Bauer C. Java Persistence with Hibernate, Second Edition / C. Bauer, G. Gavin, G. Gregory.. — 608 с. — (Manning).