



SPRING FRAMEWORK

Katedra Mikroelektroniki i Technik Informatycznych
Politechniki Łódzkiej
ul. Wólczanska 221/223 budynek B18,
90-924 Łódź

dr inż. Jakub Chłapiński

1.

Wprowadzenie

- ❑ Co to jest aplikacja internetowa?
- ❑ Co to jest szkielet aplikacyjny?
- ❑ Platformy do aplikacji internetowych
- ❑ Najpopularniejsze szkielety aplikacyjne
- ❑ Jak wybrać odpowiednią platformę/szkielet do danej aplikacji internetowej?
- ❑ Jakie są zalety platformy JavaEE?

Co to jest aplikacja internetowa?

- Program o architekturze klient-serwer
- Komunikacja pomiędzy klientem a serwerem poprzez protokoły HTTP (HTTPS)
- Interfejs użytkownika wykonany w standardzie HTML (XHTML) dostępny przy użyciu przeglądarki internetowej



- Oprogramowanie udostępniające zestaw komponentów implementujących najczęściej wykorzystywane w budowaniu aplikacji elementy
- Zawiera elementy o charakterze uniwersalnym dla większości aplikacji, nadające się przy niewielkim nakładzie pracy do wielokrotnego wykorzystania, np.:
 - Mechanizmy sesji, kontynuacji, itd.
 - Mechanizmy dostępu do źródeł danych
 - Szkielet wzorca projektowego *Model-View-Controller* lub *Model-Presenter-Controller*
 - Mechanizmy uwierzytelniania i bezpieczeństwa
- Za założenia ma ułatwiać implementowanie oraz utrzymanie aplikacji
 - Zasada wielokrotnego wykorzystywania kodu (*code reuse*)
 - Sprecyzowanie reguł projektowania aplikacji upraszcza utrzymanie kodu
- Szkielety aplikacyjne mogą wspierać większość lub tylko wybrane aspekty implementacji i bardzo często stosuje się kilka szkieletów do jednej aplikacji

- Do implementacji aplikacji internetowej można zastosować cały szereg technologii (platform), wykorzystujących różne języki programowania
 - PHP
 - JavaEE (Java)
 - ASP .NET (C#, VisualBasic)
 - Perl
 - Python
 - Ruby
 - ColdFusion (CFML)

- Nie można jednoznacznie określić który szkielet aplikacyjny lub która platforma jest najpopularniejsza.
- Pewne pojęcie o popularności danego rozwiązania daje liczba wyszukiwań w wyszukiwarkach, liczba odnośników do strony prezentującej dane rozwiązanie, liczba pobrań kodu itd.
- Według różnych źródeł najczęściej stosowane platformy to PHP, ASP .NET, Ruby, Java (JavaEE)
- Ważną pozycję na rynku posiada również Adobe Flex

Źródło: www.hotframeworks.com z dnia 2011-02-12

No	Framework	Language	Popularity Score
1	Zend	PHP	99
2	ASP.NET	C#	96
3	Ruby on Rails	Ruby	89
4	CodeIgniter	PHP	87
5	Django	Python	85
6	Symfony	PHP	81
7	CakePHP	PHP	79
8	Google Web Toolkit	Java	75
9	Yii	PHP	73
10	Flex	ActionScript	72
11	Spring	Java	71
12	ASP.NET MVC	C#	69
13	Struts	Java	69
14	Cocoon	Java	66
15	Zope	Python	66
16	Flask	Python	65
17	Sinatra	Ruby	63
18	Wicket	Java	63
19	Cappuccino	Objective-J	62
20	Express	JavaScript	62

- Dla platformy JavaEE najpopularniejsze szkielety aplikacyjne to GWT, Spring oraz Struts
- Dużą popularnością cieszy się również JSF (wg niektórych źródeł największą)
- Popularność pozostałych wymienionych szkieletów jest jednak bardzo zbliżona, wyniki niezależnych rankingów różnią się bardzo znacząco

Źródło: www.hotframeworks.com z dnia 2011-02-12

No	Framework	Language	Popularity Score
1	Google Web Toolkit	Java	75
2	Spring	Java	71
3	Struts	Java	69
4	Cocoon	Java	66
5	Wicket	Java	63
6	JSF	Java	60
7	Tapestry	Java	59
8	Seam	Java	56
9	Apache Click	Java	54
10	Play	Java	53

- Z reguły przy wyborze szkieletu aplikacyjnego lub platformy architektki muszą odpowiedzieć sobie na kilka pytań:
 - ▣ Zastosowanie którego szkieletu/platformy umożliwi zrealizowanie wszystkich założeń aplikacji?
 - ▣ Jaki szkielet/platforma pozwoli zaimplementować daną aplikację najmniejszym kosztem?
 - Jaki jest koszt wdrożenia zespołu programistów do stosowania danej technologii?
 - Jaka jest dostępność oraz jakość dokumentacji
 - Czy dana technologia jest nadal aktywnie rozwijana?
 - Jaki jest koszt (licencja) samego szkieletu/platformy?
 - Jaki jest koszt sprzętu niezbędnego do spełnienia wymagań wydajnościowych aplikacji?
 - ▣ Jakie są możliwości skalowania (wydajność) określonej platformy/szkieletu?

- Zastosowanie którego szkieletu/platformy umożliwi zrealizowanie wszystkich założeń aplikacji?
 - Można zaryzykować stwierdzenie, iż wszystkie rozwiązania umożliwiają zaimplementowanie dowolnej funkcjonalności, o ile jest to tylko możliwe wobec ograniczeń samego protokołu komunikacji HTTP, możliwości przeglądarek itd.
 - Niektóre funkcjonalności są lepiej lub gorzej wspierane przez różne szkielety aplikacyjne (np. kontynuacja, uwierzytelnianie, itd.)
 - Wniosek – najlepiej jest wybrać rozwiązanie które najlepiej wspiera najtrudniejsze (najbardziej kosztowne) do zaimplementowania funkcjonalności danej aplikacji



- Jaki szkielet/platforma pozwoli zaimplementować daną aplikację najmniejszym kosztem?
 - ▣ Najczęściej dominującą składową całkowitego kosztu są nakłady potrzebne na opłacenie zespołu analityków, architektów, programistów oraz testerów
 - Aby ograniczyć koszt należy co do zasady wybierać rozwiązania już sprawdzone i znane w zespole
 - Należy unikać rozwiązań przestarzałych i nie rozwijanych lub nie posiadających dobrze zredagowanej i kompletnej dokumentacji
 - ▣ W przypadku mniejszych aplikacji cena licencji na technologie oraz narzędzia (np. ASP .NET wraz z Windows Server oraz Visual Studio) może zniechęcać do ich wykorzystania, jednak w dużych aplikacjach najczęściej koszt ten nie ma dużego znaczenia

- Jakie są możliwości skalowania (wydajność) określonej platformy/szkieletu?
 - Popularne platformy i szkielety oferują dość zbliżoną wydajność, trudno wskazać tutaj jednoznacznego lidera
 - Pewne technologie ułatwiają implementację niektórych funkcjonalności kosztem wydajności (np. JSF oferuje przeciętnie około 3-krotnie gorszą wydajność niż JSP), jednak zmniejszenie kosztu implementacji bardzo często przewyższa koszt zakupu wydajniejszego serwera który pozwoli spełnić założone normy jakościowe.
 - Skalowalność aplikacji to sztuka sama w sobie, którą realizuje się poprzez zaprojektowanie odpowiedniej architektury aplikacji, a nie poprzez wykorzystanie konkretnej platformy lub szkieletu aplikacyjnego.
 - Można w uproszczeniu założyć, iż żadna platforma/szkielet nie skaluje się gorzej od pozostałych.

2.

Platforma JavaEE

- ❑ Zalety platformy JavaEE
- ❑ Przegląd wybranych szkieletów aplikacyjnych dla JavaEE

- ❑ Prostota i popularność języka Java
- ❑ Bardzo dobre wsparcie dla najczęściej wykorzystywanych algorytmów oraz struktur danych
- ❑ Możliwość stosowania adnotacji w kodzie Java
- ❑ Metaprogramowanie dzięki Java Reflection API
- ❑ Programowanie aspektowe
- ❑ Darmowe narzędzia o bardzo bogatych możliwościach do projektowania i implementacji aplikacji internetowych
 - ❑ Eclipse IDE for Java EE Developers (www.eclipse.org) 
 - ❑ NetBeans IDE (www.netbeans.org) 
- ❑ Dostępność platformy na wszystkich popularnych systemach operacyjnych



- Dostępność darmowych, intensywnie rozwijanych technologii oraz szkieletów aplikacyjnych do zrealizowania i wdrożenia kompletnej aplikacji internetowej
 - ▣ Serwer aplikacji (servlet-ów) (Tomcat, JBoss, Glassfish)
 - ▣ Kontener odwróconego sterowania (Spring)
 - ▣ Dostęp do baz danych (JDBC)
 - ▣ Mapowanie relacyjno-obiektowe (Hibernate)
 - ▣ Generowanie dynamicznych widoków HTML (JSP, JSF)
 - ▣ Generowanie raportów (JasperReports)
 - ▣ Wzorzec Model-View-Controller (Spring MVC, Struts 2)
 - ▣ Wsparcie dla testów jednostkowych (JUnit)
 - ▣ i wiele innych...