

Aplikacje webowe według wzorca MVC

Adam S. Grzonkowski

245594, IS III rok

12 maj 2015

Podsumowanie

- ▶ Wzorce architektoniczne i projektowe
 - ▶ Czym są, do czego mogą się przydać
- ▶ MVC – krótkie omówienie
 - ▶ Model
 - ▶ View
 - ▶ Controller
- ▶ MVC – jak to działa
 - ▶ Omówienie sposobu działania
- ▶ Przykład

Wzorce architektoniczne

- ▶ Uznany i sprawdzony sposób rozwiązania danego problemu z zakresu architektury oprogramowania.
- ▶ Wzorce architektoniczne określają ogólną strukturę danego systemu informatycznego:
 - ▶ elementy z jakich się składa,
 - ▶ zakres funkcji realizowanych przez dany element,
 - ▶ zasady komunikacji pomiędzy poszczególnymi elementami.

Wzorce projektowe

- ▶ Uniwersalne,
 - ▶ sprawdzone w praktyce,
 - ▶ rozwiązanie często pojawiających się,
 - ▶ powtarzalnych problemów projektowych.
-
- ▶ Pokazuje powiązania i zależności pomiędzy klasami oraz obiektami i ułatwia tworzenie, modyfikację oraz pielęgnację kodu źródłowego. Wzorce projektowe stosowane są w projektach wykorzystujących programowanie obiektowe.

Wzorce projektowe

- ▶ UWAGA!
- ▶ Wzorzec jest opisem rozwiązania, a nie jego implementacją.
- ▶ Pojęcie istnieje od 1987 roku (Kent Beck i Ward Cunningham),
- ▶ spopularyzowane dopiero w 1995 roku dzięki książce *Inżynieria oprogramowania: Wzorce projektowe*.

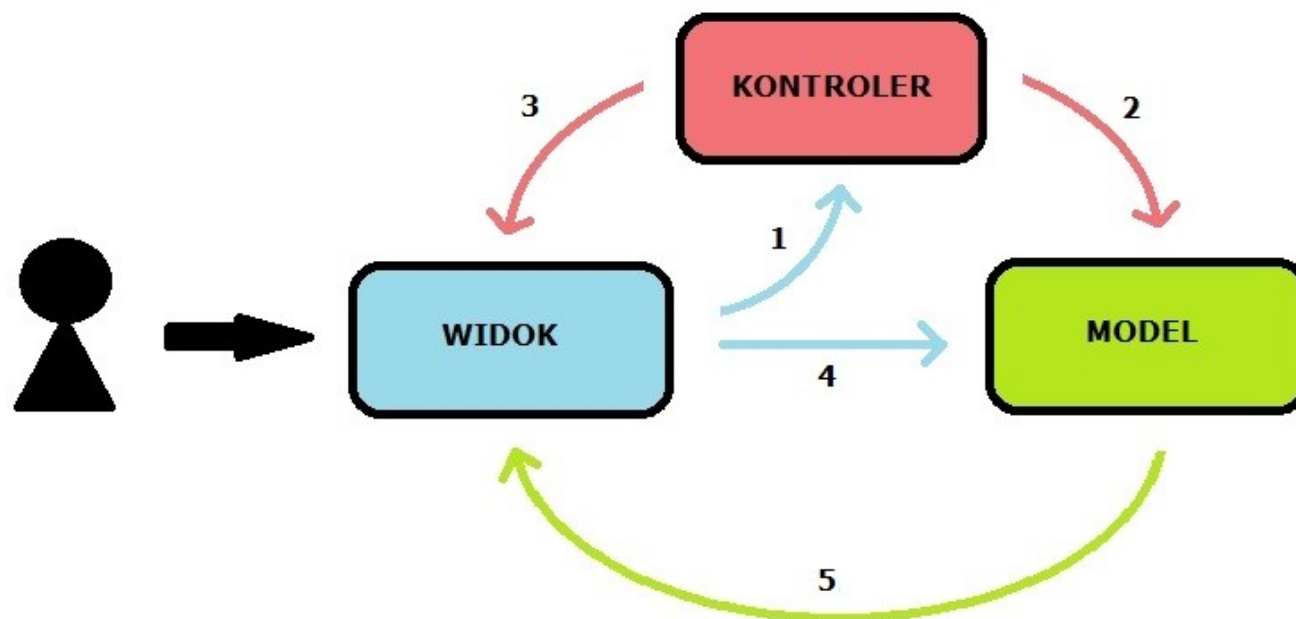
MVC – zarys historyczny

- ▶ Koncepcja zaprezentowana przez Tryvge Reenskaug w latach 1970-ych
- ▶ Zaimplementowana w latach 1980-ych przez Jima Althoffa
- ▶ Sukcesywnie popularyzowana od 1988 roku, dzięki artykułowi The Journal of Object Technology
- ▶ MVC był pierwszym wzorcem architektonicznym swojego rodzaju – tj. pomagającym szybko utworzyć graficzny interfejs użytkownika dla aplikacji webowych.

MVC – co to jest

- ▶ Model – reprezentacja logiki aplikacji.
- ▶ View – wyświetlanie części modelu w ramach interfejsu użytkownika.
- ▶ Controller – przyjmuje dane wejściowe od usera, reagując np. aktualizacją modelu lub odświeżeniem widoku.
- ▶ MVC to wzorzec architektoniczny, stosowany obecnie najczęściej przy tworzeniu aplikacji webowych.
- ▶ Trzy zbliżone wzorce: Model View Presenter, Model View View Model, Model View Adapter.

MVC – jak działa



MVC – jak działa

- ▶ Użytkownik klika element na stronie opartej o MVC:
- ▶ 1. Żądanie wyświetlenia wybranej aktualności przesyłane jest z *widoku* do *kontrolera*, którego celem jest wybranie odpowiedniego *modelu* i *widoku*.
- ▶ 2. *Kontroler* po rozpoznaniu żądania wybiera *model* reprezentujący aktualność.
- ▶ 3. *Kontroler* wybiera *widok* do prezentacji aktualności i przekazuje mu informacje o *modelu*, z którego ma skorzystać.
- ▶ 4. *Widok* wysyła zapytanie do *modelu* w celu pobrania treści aktualności.
- ▶ 5. *Model*, w którym wykonuje się logika biznesowa, posiada wszystkie potrzebne dane od *widoku*, więc spełnia jego żądanie. Znajduje odpowiedni wpis w bazie i przesyła jego treść.

Przykład

- ▶ Struktura katalogów MVC
 - ▶ Krótko o tym co gdzie znaleźć
- ▶ Omówienie modelu bazy danych
 - ▶ Omówienie stworzonego wcześniej modelu – co w nim jest
- ▶ Stworzenie bazy danych
 - ▶ Stworzenie bazy danych na potrzeby naszej aplikacji
- ▶ Stworzenie kontrolera
 - ▶ Kontroler będzie obsługiwać komunikację między widokiem, a modelem
- ▶ Stworzenie widoków
 - ▶ Stworzenie nowych widoków dla naszej bazy danych

Różnice między wersjami

ASP.NET MVC3	ASP.NET MVC4	ASP.NET MVC5
▪ Templates for HTML 5 and CSS 3 ▪ Improved Model validation ▪ Razor View Engine ▪ Support for Multiple View Engines ▪ Controller improvements ▪ Unobtrusive <i>JavaScript</i> approach ▪ Improved Dependency Injection ▪ Partial page output caching http://www.webdevelopmenthelp.net	▪ ASP.NET Web API ▪ Adaptive rendering ▪ Look-n-Feel improvements ▪ Empty Project Template. ▪ Mobile Project Template ▪ Support for adding controller ▪ Task Support for Asyn Controllers ▪ Bundling and Minification ▪ Support for OAuth and OpenID ▪ Support for <i>Windows Azure SDK 1.6</i>	▪ ASP.NET Identity ▪ Authentication Filters ▪ Filter overrides ▪ Bootstrap ▪ Attribute Routing

Dziękuję za uwagę