Projektowanie aplikacji ASP.NET
Zestaw 3

Web Forms Data Binding

08-11-2016

Liczba punktów do zdobycia: **10/30**
Zestaw ważny do: 22-11-2016

1. (**1p**) Związ imperatywnie (w kodzie C#) dowolny formant wiązalny, np. `DropDownList` z relacyjnym źródłem danych (dowolny `IListSource` np. `DataSet, DataTable, DataView`) a potem z obiektowym (dowolny `IEnumerable`, np. `ArrayList, List<T>`).

   Jak określać formatowanie wartości dla typów prostych na poziomie deklaratywnym (przykład: na liście rozwijalnej mają pokazać się wartości typu `DateTime` z formatowaniem `"d"`?

   Jak określać formatowanie wartości dla typów referencyjnych na poziomie deklaratywnym (przykład: obiekt implementuje interfejs `IFormattable` i chcę żeby na liście rozwijalnej pokazał się z formatowaniem `"d"`?

   Wskazówka: `DataTextField`, `DataTextFormatString` *oraz pokazany na wykładzie sposób z propercją* `This` *w klasie.*

2. (**1p**) Zademonstrować jakikolwiek inny niż `ObjectDataSource` obiekt dostarczający dane do wiązania, np. `SqlDataSource`, `XmlDataSource`, `LinqDataSource`.

3. (**4p+2p+2p**) **(Sklep internetowy)**

   Zademonstruj w praktyce składanie warstwy prezentacji opartej na formancie `ListView` z warstwą danych opartą na komponencie `ObjectDataSource`. Posłuż się przykładami z wykładu aby szybko osiągnąć efekt prezentacji danych i tylko dodawaj kolejne funkcjonalności.

   Prezentowany widok ma dotyczyć towarów w sklepie internetowym - każdy towar ma mieć nazwę, opis, cenę i link do obrazka. Zadbaj więc o to, żeby `ListView` miał właściwie opisany szablon elementu (`ItemTemplate`).

   Zwróć uwagę na poprawne przekazywanie parametrów **sortowania** i **wybierania strony** danych do warstwy mapowania obiektowo-relacyjnego w taki sposób, żeby zapytanie wysyłane ostatecznie do serwera bazy danych było optymalne (tzn. wybierało faktycznie **stronę** danych w zadanym **porządku**).

   Formant powinien obsługiwać **cztery** rodzaje operacji - prezentację, modyfikację, dodawanie i usuwanie. Innymi słowy - widok ma charakter administracyjny, administrator sklepu internetowego nie tylko przegląda listę towarów ale też może każdy z nich wyedytować oraz dodawać nowe.

   Dwa wydzielone punkt za osobno zaimplementowany koszyk (cart) - każdy towar można dodać do koszyka z poziomu widoku towarów, a następnie przejść do osobnego widoku, w

którym widać tylko te elementy które wcześniej dodano do koszyka. Koszyk zaimplementować jako osobną klasę `ShoppingCart`, która wewnętrznie używa kontenera `Session` do tymczasowego przechowania wybranych towarów na czas sesji pracy użytkownika.

Dwa wydzielone punkty za `ObjectDataSource` który odczytuje dane z jakiegoś rzeczywistego systemu bazodanowego (vs obsługa statycznych list w pamięci). Najszybciej będzie oczywiście użyć technologii oferującej zapytania LINQ. Warto zapamiętać zaprezentowaną na wykładzie technologię Dynamic LINQ, ułatwiającą obsługę porządkowania wyników zapytań.

Wiktor Zychla