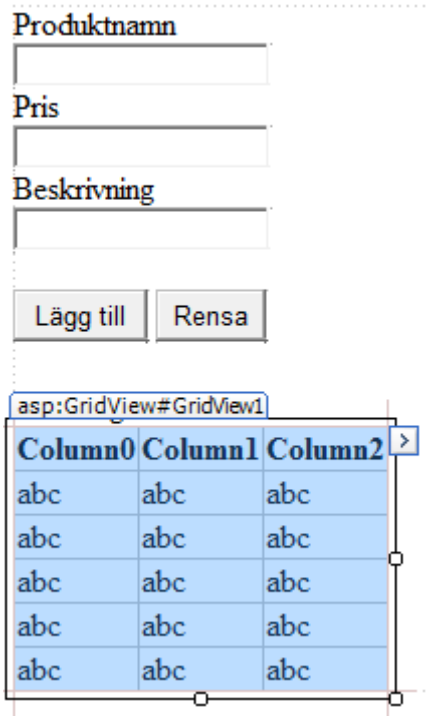
LABORATION 4

Skapa en kundvagn baserad på två olika klasser

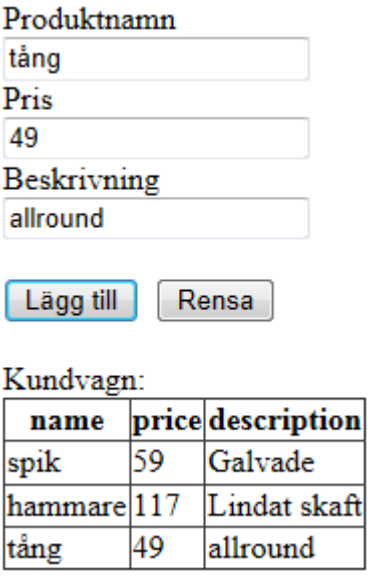
Laborationen innebär att ni ska skapa en kundvagn för att öva ytterligare på objektorienterad programmering med egna klasser. Som vanligt skapar ni klassfiler med ändelsen .cs som ni placerar i App_Code-mappen. Ni använder er av Visual Studio 2010. Kom ihåg att vissa saker ni gör under laborationen ska redovisas i samband med muntan. Spara hela tiden underlaget på det ni gör (kopiera hela projektmappen så ni garanterat har med alla filer då ni sparar).

Arbetsuppgift 1 – Skapa gränssnitt

Vi börjar med att skapa gränssnittet. Starta VS2010 och skapa ett tomt webbprojekt. Skapa en ny webbsida genom att högerklicka på projektet och lägg till en default.aspx sidan (code-behind sidan genereras automatiskt). Vi ställer oss i desinvyn och drar ut labels och textboxar så gränssnittet ser ut ungefär som det gör i figur 1. Vi behöver dessutom ett presentationsobjekt, t ex en gridview, som är markerad i figur 2 (den visar resultatet då ni lägger till produkterna).



Figur 1. Skapa gränssnitt



Figur 2. Utseende då ni lagt till några produkter i kundvagnen

Arbetsuppgift 2 – skapa funktionalitet och klassfiler

Skapa funktionalitet bakom knapparna i codebehind-filen, se default.aspx.cs. Dubbelklicka på respektive knapp för att skriva in kod för klickhändelserna (då användaren klickar på respektive knapp).

Knappen ”lägga till” ska lägga till en produkt i kundkorgen. Kundkorgens innehåll ska därefter visas i er gridview så vi hela tiden ser innehållet i den. Ni skapar (deklarerar) sträng-variabler för att lagra textboxarnas innehåll i. Ni använder klassen produkt för att lägga till ett produktobjekt (variablerna ni skapat innan används som indata till konstruktorn), se product.cs som ledning för hur ni kan konstruera klassen. När ni skapat produkten kan ni direkt efter lägga till den i er shoppingcart. Ni använder då produkten som indata för att via metoden addProduct() lägga till produkten i en arraylist inne i klassen shoppingcart, se ShoppingCart.cs som ledning för hur ni kan konstruera ShoppingCart-klassen. Arraylistens add() funktion används för att lägga till produkter med. Tänk på att klassfilerna ska placeras i App_Code-mappen.

Knappen ”rensa” ska tömma vår shoppingcart. Det finns i en arraylist (som används inne i klassen ShoppingCart) en färdig metod för att rensa allt som är inlagt som heter clear(), se metoden emptyCart() i shoppingcart-klassen.

För visning av innehållet i vår gridview behöver vi skapa en metod som returnerar innehållet i den som tex heter getProducts(). Den returnerar enbart innehållet i den interna ArrayList som finns inne i klassen shoppingcart.

Sammanfattningsvis kan man säga att Ni skapar en shoppingcart som därefter innehåller alla de produkter ni lägger till i den. Shoppingcarten är static så den kommer ihåg de produkter ni lägger till i den (då ni laddar om sidan och då tappar den värdena i annat fall).

Kommentera er kod i början av subrutiner eller där det behövs så det är lätt att förstå vad som händer inne i er kod (ej rad för rad-kommentering). Kommer ni tillbaka till koden efter någon vecka eller månad ska det vara lätt att förstå den. Kommetarer skrivs med ”//” åtföljt av kommentaren.

Arbetsuppgift 3

Dokumentera ert system med hjälp av UML. Rita upp klassen samt aktivitetsdiagram för något av den logik ni skapat med kod (se under föreläsning utdelat häfte över UML-notation). Ni kan arbeta med vanligt papper och penna eller om ni vill arbeta digitalt tex i Excel… Dessa beskrivningar samt kodutdrag ska bifogas i samband med muntan.

Lycka till!