|  |  |
| --- | --- |
| **Øvelse** | **ButikV2** |
| **Project** | **ButikV2** |
| **Formål** | Få øvelse i at arbejde med **List**-klassen |
| **Description** | I Visual Studio-solutionen **ButikV2** er der fra starten givet to klasser **Kunde** og **Produkt**. Vi skal i øvelsen arbejde med lister af hhv. **Kunde**-objekter og **Produkt**-objekter. |
| **Steps** | 1. Vær sikker på, at du har ”unzippet” den fil, som **ButikV2** ligger i (VIGTIGT). 2. Åbn **ButikV2** i Visual Studio, og se lidt på klasserne **Kunde** og **Produkt**. De er ikke specielt komplicerede, men det er vigtigt at vide hvad de rummer af properties og metoder. 3. I **Program.cs** laves der en liste af **Kunde**-objekter. Herefter følger der tre opgaver (angivet som kommentarer i **Program.cs**) der skal løses for denne liste. Løs disse opgaver. 4. Derefter laves der en liste af **Produkt**-objekter. Herefter følger der tre opgaver (angivet som kommentarer i **Program.cs**) der skal løses for denne liste. Løs disse opgaver. 5. Hvis du stadig har tid tilbage, kan du overveje disse **noget sværere** skridt:    1. Lav en **Ordre**-klasse, som skal rumme en reference til et **Produkt**-objekt, en **Antal**-property, samt en beregnet property **SamletPris**. Et **Ordre**-objekt skal således repræsentere, at man har bestilt **Antal** styk af **Produkt**.    2. Udvid **Kunde**-klassen, så den nu rummer en liste af **Ordre**-objekter. Dermed skal property’en **AntalOrdrer** nu være beregnet. Du får også brug for at udvide **Kunde**-klassen med f.eks. en metode til at tilføje en **Ordre** til en **Kunde**. Med dette på plads skal der også tilføjes en property **SamletPrisForOrdrer** til **Kunde**-klassen.    3. Sæt et antal **Kunde**-, **Produkt**- og **Ordre**-objekter op i **Program.cs**, og se om du f.eks. kan beregne den totale pris for alle kundernes ordrer.    4. Find selv på yderligere udfordringer 😊. |