|  |  |
| --- | --- |
| **Øvelse** | SpeedRun06 |
| **Project** | **SpeedRun6** |
| **Formål** | Implementér en klasse fra bunden |
| **Beskrivelse** | I Visual Studio-solutionen **SpeedRun6** er der fra starten en tom klasse-definition af klassen **Item**. Denne klasse-definition skal du nu implementere, så den opfylder neden­stående krav (se under Trin).  **NB**: Bemærk, at **SpeedRun6** ikke kan køre fra starten, da der er kode i **Program.cs** som først kan køre når klassen **Item** er korrekt imple­men­­teret.  NB: **SpeedRun6** rummer også en klasse kaldet **ItemRepository**, som kun skal bruges, hvis man når til Trin 5. |
| **Trin** | 1. **NB**: Vær sikker på, at du har ”unzippet” den fil, som **SpeedRun6** ligger i. 2. Åbn **SpeedRun6** i Visual Studio. Den rummer en tom klasse-defini­tion af klas­sen **Item**. Der stilles nu følgende krav til klassen **Item**:    1. Alle properties skal have både en **get** og en **set**.    2. Den skal have en property kaldet **Id**, som skal rumme et heltalligt løbe-nummer for et **Item**, f.eks. 2 eller 14.    3. Den skal have en property kaldet N**ame**, som skal rumme navnet for et **Item**, f.eks. ”Keyboard” eller ”Monitor” .    4. Den skal have en property kaldet **Price**, som angiver prisen for et **Item**.    5. Den skal have en constructor, som tager parametre til at initialisere de ovenstående properties. 3. Afprøv den færdige **Item**-klasse ved at prøve at køre **SpeedRun6**. Så længe du ikke kan køre **SpeedRun6**, er du ikke færdig med dette trin. 4. Når **SpeedRun6** kører, kan du ved hjælp af udskrifterne på skærmen også se, om alt virker som det skal. Hvis det ikke er tilfældet, må du finde og rette fejlene. 5. **EKSTRA** (kun hvis du er hurtigt færdig med Trin 1-4). **SpeedRun6** rummer også klassen **ItemRepository**, som kan referere til et antal **Item**-objekter (se koden for **ItemRepository**). Der er også tilføjet kode i **Program.cs** som afprøver **ItemRepo-sitory** ift. de nedenstående trin. Du skal implementere følgende:    1. Den (ufærdige) beregnede property **TotalPrice** skal returnere den samlede pris af alle de **Item**-objekter **ItemRepository** rummer.    2. Den (ufærdige) metode **PrintAll** skal udskrive alle de **Item**-objekter **Item-Repository** rummer (tip: måske kunne det være bekvemt, hvis du imple-menterede en **ToString**-metode for **Item** først….)    3. **ItemRepository** gemmer **Item**-objekter i en **Dictionary**. Overvej – og kod det også gerne – hvordan de forskellige elementer i **ItemRepository** vil ændre sig, hvis vi beslutter at bruge en **List** i stedet for. |