|  |  |
| --- | --- |
| **Øvelse** | SpeedRun03 |
| **Project** | **SpeedRun3** |
| **Formål** | Implementér en klasse fra bunden |
| **Beskrivelse** | I Visual Studio-solutionen **SpeedRun3** er der fra starten en tom klasse-definition af klassen **Kunde**. Denne klasse-definition skal du nu implementere, så den opfylder nedenstående krav (se under Trin).  **NB**: Bemærk, at **SpeedRun3** ikke kan køre fra starten, da der er kode i **Program.cs** som først kan køre når klassen **Kunde** er korrekt imple­men­­teret.  NB: **SpeedRun3** rummer også en klasse kaldet **Vare**, som kun skal bruges, hvis man når til Trin 5. |
| **Trin** | 1. **NB**: Vær sikker på, at du har ”unzippet” den fil, som **SpeedRun3** ligger i. 2. Åbn **SpeedRun3** i Visual Studio. Den rummer en tom klasse-definition af klas­sen **Kunde**. Der stilles nu følgende krav til klassen **Kunde**:    1. Den skal have en property kaldet **Navn**, som skal rumme nav­net på kunden, f.eks. ”Per Jensen” eller ”Mona Thomsen”. Værdien af **Navn** kan ikke ændres efter at et **Kunde**-objekt er blevet oprettet.    2. Den skal have en property kaldet **Email**, som skal rumme e-mail adressen for kunden, f.eks. ”peje@mail.dk”. Værdien af **Email** kan godt ændres efter at et **Kunde**-objekt er blevet oprettet.    3. Den skal have en property kaldet **VIP**, som angiver om kunden er en VIP-kunde. Værdien af **VIP** kan godt ændres efter at et **Kunde**-objekt er blevet oprettet, men starter for alle kunder med at være *false*.    4. Den skal have en constructor, som tager parametre til at initialisere **Navn** og **Email**, og som initialiserer **VIP** til *false*. 3. Afprøv den færdige **Kunde**-klasse ved at prøve at køre **SpeedRun3**. Så længe du ikke kan køre **SpeedRun3**, er du ikke færdig med dette trin. 4. Når du kan køre **SpeedRun3**, kan du ved hjælp af udskrifterne på skærmen også se, om alt virker som det skal. Hvis det ikke er tifældet, må du finde og rette fejlene. 5. **EKSTRA** (kun hvis du er hurtigt færdig med Trin 1-4). Lav nu disse ændringer og tilføjelser til **Kunde**:    1. Man skal nu kunne knytte en kunde sammen med et antal varer, d.v.s. et **Kunde**-objekt skal kunne referere til et antal **Vare**-objekter. Find en måde at implementere dette på (Hint: der skal tilføjes noget til **Kunde**-klassen, som kan rumme et antal **Vare**-objekter).    2. Man skal også på et givet **Kunde**-objekt nemt kunne få svar på, hvad den samlede pris er for kundens varer. Implementér dette som en beregnet property i **Kunde**-klassen, og afprøv **i Program.cs**. |