# Disaheim 3

|  |  |
| --- | --- |
| **Læringsmål** | Du kan:   * **1Pf1:** anvende centrale metoder til at specificere og konstruere algoritmer [...] * **1Pf2:** anvende centrale faciliteter i programmeringssproget til realisering af algoritmer, designmønstre, […], datatyper, datastrukturer, designmodeller […] * **1Pf3**: anvende et i professionen udbredt, integreret udviklingsværktøj, herunder versionsstyringssystem […] til at designe og konstruere praksisnære applikationer […] * **1Pk3**: i en struktureret sammenhæng tilegne sig ny viden, færdigheder og kompetencer inden for programmeringssprog, udviklingsværktøjer, programmeringsteknikker og programdesign * **1Pf6**: anvende moderne teknikker og værktøjer til afvikling af test […] |
| **Forventet læringsudbytte [SOLO]** | Programmeringssporet:   * [Unistrukturel] Du kan huske enkelte kendetegn ved et foreach-**loop** * [Multistrukturel] Du kan beskrive flere kendetegn kendetegn ved **C# properties** * [Multistrukturel] De kan beskrive flere kendetegn i forhold til indkapsling, polymorfi og abstraktion(**OOP-principper**) * [Unistrukturel] Du kan nævne enkelte kendetegn (bl.a. base class og derived class) ved **nedarvning (en: inheritance)** (OOP-princip) * [Multistrukturel] Du kan beskrive og anvende flere kendetegn (bl.a. brug af this/base, overloading, mm.) ved en constructor i en **C# klasse - Constructor** * [Unistrukturel] Du kan huske enkelte kendetegn ved et class library |
| **Din forberedelse** | Programmeringssporet:   * [C# foreach loop - Tutorial for beginners](https://www.youtube.com/watch?v=k_SmGu7GYYU) (video: 8:42) * [C# foreach loop](https://www.codebuns.com/csharp-basics/foreach-loop/) * [Inheritance in C# and .NET](https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/inheritance) (skimlæs: lidt lang tutorial, men dækker stort set alle begreber om både nedarvning og abstrakte klasser/metoder) * [Understanding inheritance](https://channel9.msdn.com/Series/C-Fundamentals-for-Absolute-Beginners/17) (video: 33:08)   + Bemærk at msdn.microsoft.com nu ligger på docs.microsoft.com, og at du bruger .NET 5, og ikke .NET Framework, som Bob Tabor viser * [Understanding abstract Classes and abstract methods](https://www.linkedin.com/learning/learning-c-sharp-3/understanding-abstract-classes) (video: 5:04) * [Static Classes and Static Class Members](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/static-classes-and-static-class-members) * [Adding a Class Library](https://www.youtube.com/watch?v=Jms3C-J-m2M) (video: 9:22) |

Du fortsætter igen med Disaheim-casen, hvor du får mulighed for at anvende den sidste løkke-konstruktion foreach-løkken til din implementering.

Vigtigst er dog, at du nu introduceres til det objekt-orienterede princip **nedarvning (en: inheritance)**; et centralt princip, som du vender tilbage til igen og igen, i forhold til at sammensætte og relatere klasser, så de er så enkle og nemme at vedligeholde som muligt. Det leder også frem til et tema, som du snart kommer til at stifte bekendtskab med, nemlig fordeling af ansvar mellem klasser (GRASP). Opgaven introducerer dig også til abstrakte klasser og abstrakte metoder, som hænger tæt sammen med nedarvning samt til statiske klasser og statiske klassemedlemmer.

Desuden skal du se på, hvordan du laver dit eget bibliotek i C# (Class Library).

# Dagens ord:

Always code as if the guy who ends up maintaining your code   
will be a violent psychopath who knows where you live.   
(Martin Golding)