# Bank Account and Sorting

|  |  |
| --- | --- |
| **Læringsmål** | Du kan:   * **1Pf1:** anvende centrale metoder til at specificere og konstruere algoritmer [...] * **1Pf2:** anvende centrale faciliteter i programmeringssproget til realisering af algoritmer […] * **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 styring af **program flowet** – sekvens, if-else, switch, while, do-while, for, foreach, metodekald og -retur, exception handling, jump statements * [Unistrukturel] Du kan genkende enkelte kendetegn ved simple **C#** **datatyper** - int, string, bool, double, char, array (Sort) * [Unistrukturel] Du kan genkende enkelte kendetegn ved **C# klasse (objekt)** –  […], property, backing field, get, set, auto-implemented properties   Færdighedssporet:   * [Unistrukturel] Du kan nævne enkelte kendetegn ved **testing** – strukturen af en unit test i Visual Studio IDE (Test class attributes (TestClass, TestMethod, TestInitialize), assertions (Assert, CollectionAssert) * [Unistrukturel] Du kan huske enkelte kendetegn ved **debugging** –break point, step into, step over |
| **Din forberedelse** | Programmeringssporet:   * [Properties](https://csharp.net-tutorials.com/classes/properties/) * [Sort Array in C# without using inbuilt function](https://www.c-sharpcorner.com/blogs/sort-array-in-c-sharp-without-using-inbuilt-function) * [Using foreach with arrays](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/using-foreach-with-arrays) * [How to override the ToString method](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/how-to-override-the-tostring-method) * Properties – auto-implemented & side effects.mp4 (video: 29:17; se i Planer i ItsLearning)) |

# Dagens ord: Forløse frustration og forståelse af flow

Du har måske oplevet en eller flere af de følgende frustrationspunkter:

* ”Hjælp, jeg forstår ikke, hvorfor min metode ikke returnerer det resultat, som jeg forventede”
* ”Hjælp, min løkke virker ikke, som den skal”
* ”Hjælp, jeg kan ikke forstå den kode, som vi har udarbejdet i gruppen”
* ”Hjælp, jeg kan ikke forstå den kode, som jeg har fået udleveret”
* ”Hjælp, jeg kan ikke forstå den kode, som jeg har fundet på [stackoverflow](https://stackoverflow.com/)”

Et værktøj, som du kan benytte til at forløse denne frustration og til at opnå en personlige forståelse af selve programflowet, er: **Debugging**

***Hvad så med unit tests?***

Ja, hvad så, hvis du har modtaget en række af unit tests, og du kan se, at din implementering fejler, og at du ikke modtager det forventede resultat? Kan du så benytte debugging her? Ja! Åbn tester exploreren, højreklik på den unittest, der fejler, og vælg ”Debug”, som på billedet nedenunder:

