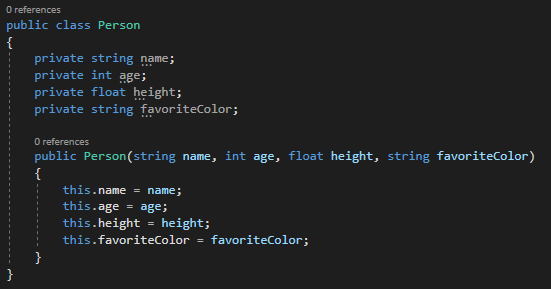
**Refleksion af første uge på H2 (Objektorienteret prog.)**

**Classes / Libraries**

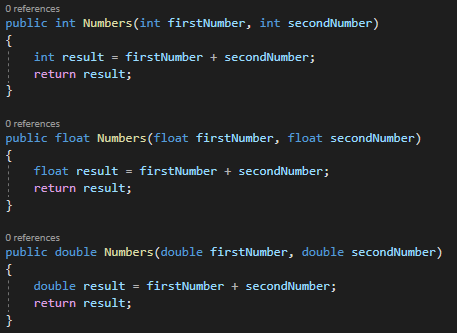
* Klasser er noget der ofte bliver brugt til at lave objekter. Det består af nogle variabler og sommetider metoder. Variablerne bliver brugt til at beskrive objektet og metoderne er til at give objektet funktion, altså hvad objektet gør. Constructors er også en del af objekter. Det gør at man kan sætte parametre ind, så man kan lave om på variablerne, hver gang man laver et nyt objekt.

**Dictionary**

* Dictionaries er ligesom en list eller et array. Det er en ting man kan gemme variabler i og hente ud derfra igen, hvis man skal bruge det. Dog i en dictionary, bliver tingene gemt med en key. Så f.eks. kan man gemme en string og så finde den med en key af type int. Så hvis vi siger at jeg gemmer ”Programmering” med key’en 69, så ville jeg kunne hente min string ved skrive key’en. Dictionaries har heller ikke nogen rækkefølge ligesom lister og arrays har, så det bliver smidt tilfældigt ind, hvilket er grunden til at man har en key, så uanset hvor den er henne, så kan man få fat ved hjælp af den.

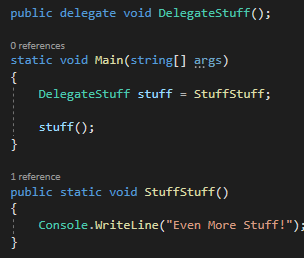
**UML Diagram**

* UML Diagrammer er en form for diagram, hvor man kan skabe sig et overblik over hvordan ens klasser skal hænge sammen. Det kan være meget brugbart hvis man kan gøre det ordenligt, så det eneste der skal gøres, er at lave klassen med de variabler og metoder man har tænkt sig og så få den til at referere til andre klasser/libraries.

**Overloading**

* Overloading er når man gemmer flere funktioner med det samme navn, men hvor den har forskellige input og retur typer, for hver funktion. Jeg vil tænke at det kunne bruges til at bruge forskellige værdier hvor det er bedst at den hedder det samme

**Delegates**

* Nu har jeg ikke forstået det her helt så godt, men ud fra hvad jeg kan se, er delegates mere kode til noget man kunne havde skrevet på mindre linjer. Det er garanteret mere funktion i det som jeg ikke ved så meget om, men håber på at kunne forstå mere af det senere i forløbet.

Det her er hvad jeg har kunnet lave uden at have fejl i det. ->