

Klassen

Een heel elementaire klasse heeft de volgende vorm:

```
class ClassName
{
    // hier komen de data en functionaliteit
}
```

In C# kunnen we geen objecten aanmaken voor we een klasse hebben gedefinieerd die de algemene eigenschappen (properties) en werking (methoden) beschrijft

Attributen

```
class Auto {
    public int Kilometers;
    public double Benzine;
    public DateTime LaatsteOnderhoud;
}
```

attributen zijn er om objecten en klassen te omschrijven

access modifiers

```
// in Program.cs
public static void DemonstreerAttributen() {
    Auto auto1 = new Auto();
    Auto auto2 = new Auto();
    auto1.Voltanken();
    auto1.Rijden(5);
    auto1.Rijden(10);
    auto1.Rijden(20);
    Console.WriteLine(auto1.Kilometers);
    Console.WriteLine(auto2.Kilometers);
}
```

Access modifiers

Enums

```
namespace Programmeren {
    enum Weekdagen
    {Maandag, Dinsdag,
    Woensdag,
    Donderdag, Vrijdag,
    Zaterdag, Zondag}
}
```

Het keyword enum geeft aan dat we een nieuw datatype maken. Gelijkaardig aan wat we voor een klasse doen, dus. Wanneer we dit nieuwe type hebben gedefinieerd kunnen we dan ook variabelen van dit nieuwe type aanmaken (dat is waarom we spreken van een "datatype"). Anders dan bij een klasse beperken we tot alleen de opgesomde opties.

Folder Aanmaken/ project Maken in folder

folder aanmaken
Ga naar Documents
select in vekenner maak een nieuw folder aan
open visual studio code en ga naar debetreffende folder
run
Dotnet new console
geen hoofdletters

objecten

```
Auto mijnEerste = new
Auto();
Auto mijnAndereAuto =
new Auto();
```

objecten

die zijn instanties van een klasse

en verder onder objecten zijn er zo van die omschrijvende karakteristieken

een bezine soort (red . Diesel of gewone bezine)
de kleur van de auto
de aantal deuren
de capaciteit van de motor (red. hoeveel 100 pks)

objecten

```
Auto mijnEerste = new
Auto();
Auto mijnAndereAuto =
new Auto();
```

objecten

die zijn instanties van een klasse

en verder onder objecten zijn er zo van die omschrijvende karakteristieken

een bezine soort (red . Diesel of gewone bezine)
de kleur van de auto
de aantal deuren
de capaciteit van de motor (red. hoeveel 100 pks)

DateTime

```
System.Threading.Thread.Sleep(1000);
DateTime today = DateTime.Now;
DateTime borodino_battle = new
DateTime(1812, 9, 7);
TimeSpan diff = today - borodino_
battle;
WriteLine("{0} days have passed
since the Battle of Borodino.",
diff.TotalDays);**
private int kilome-
**DateTime today = DateTime.Now;
bool isLeap = DateTime.IsLeapYear(
oday.Year);
if(isLeap == true) {
    Console.WriteLine("This is
a leap year");
}**
```

DateTime objecten

men can verschillende date time objecten aanmaken

de bedoeling is een Date in C# correct weer te geven

CamelCase + PascalCase

camelCase -- klassen --
methods enz
camelCase -- variables arrays and other elements

Properties

```
private double
benzine;
public double
Benzines
{
    get
    {
        return
        benzine;
    }
    set
    {
        benzine =
        value;
    }
}
```

Getting ad setting of access modifiers

Object en klassen weergeven

Static betekend niet onveranderlijk of vast

