

Let's break down this class:

# C# Class Definition

Wir schlüsseln eine Klasse auf:

```
public class ClassName
{
    // class fields and methods
}
```

- `public`
  - Gibt die Zugriffsebene der Klasse an.
  - `public` bedeutet, dass auf die Klasse von jedem anderen Code in derselben Assembly oder einer anderen Assembly aus zugegriffen werden kann.
- `class`
  - gibt an, dass dies eine Klassendefinition ist.
- `ClassName`
  - der Name der Klasse.
- `{ }`
  - der Block, in dem wir die Felder und Methoden der Klasse definieren.

## Namespace und Klassen-Definition:

```
public class ClassName
{
    private int field1;
    public string field2;
}
```

- `private`
  - Gibt an, dass auf das Feld nur innerhalb der Klasse zugegriffen werden kann.
- `public`
  - Gibt an, dass auf das Feld von außerhalb der Klasse zugegriffen werden kann.
- `int`
  - der Datentyp des Feldes.
- `string`
  - der Datentyp des Feldes.
- `namespace Lamp;`
  - Gibt den Namespace für diese Klasse Lamp an.
- `public class GreenLamp : ClassLamp`
  - erklärt eine neue öffentliche Klasse namens GreenLamp, welche die Eigenschaften von ClassLamp erbt.
- Das `double dot : Symbol`
  - Signalisiert das hier eine Vererbung vorliegt.

## Konstrukteur

- `public GreenLamp() : base("green") { }`
- ist der Konstruktor für die GreenLamp Klasse
- `public`: bedeutet, dass auf den Konstruktor von außerhalb der Klasse zugegriffen werden kann.
- `GreenLamp()`
- ist die Konstruktorsignatur, die keine Parameter akzeptiert
- `: base("green")`
- ruft den Konstruktor der Basisklasse ClassLamp auf.
- Mit dem Argument "green" ersetzen wir die color im Feld der ClassLamp Instanz zu "green".

## Statische Methode

- `public static void MgreenLamp()`

- ist eine statische Methode, die aufgerufen werden kann, ohne eine Instanz der zu erstellen GreenLamp Klasse.
- `public`
- bedeutet, dass auf die Methode von außerhalb der Klasse zugegriffen werden kann.
- `static`
- bedeutet, dass die Methode eher zur Klasse selbst als zu Instanzen der Klasse gehört.
- `void`
- bedeutet, dass die Methode keinen Wert zurückgibt.
- `MgreenLamp()`
- ist die Methodensignatur, die keine Parameter akzeptiert.

Der Methodenkörper erstellt eine neue Instanz von ClassLamp mit der Farbe "green" und ruft die TurnOn Methode drauf.

Das Interessante an dieser Klasse ist:

- Die GreenLamp Klasse erbt von der ClassLamp, Dies bedeutet, dass GreenLamp Zugriff auf alle Member und dessen Felder, Methoden und Eigenschaften hat
- Der Konstrukteur von GreenLamp setzt das Feld color der ClassLamp Instanz zu "green",
- jede Instanz von GreenLamp wird jetzt ein Feld color "green" besitzen.
- Die Methode MgreenLamp erstellt eine neue Instanz von ClassLamp mit der Farbe "green"
  - und ruft die TurnOn Methode auf