

	Klasse	Information	Thema Ein-/Ausgabe mit Streams	
	Lehrer Oliver Kreuer	Lernfeld 6	Datum	Name

## Konstruktor und Destruktor einer Klasse

Für jede beliebige Klasse kann man im public-Bereich zwei besondere Methoden definieren:

1. Konstruktor
2. Destruktor

Für die folgenden Beispiele wird eine fiktive Klasse „Fahrzeug“ verwendet, die als einzige Variable die Fahrzeugmarke enthält:

```
class Fahrzeug
{
    char *marke[20];
};
```

### 1. Konstruktor

Prototyp: `Fahrzeug();`

Implementation:

```
Fahrzeug::Fahrzeug()
{
    strcpy(marke, ""); // Variable „marke“ mit
                        // Leerstring
                        // initialisieren.
}
```

- Ein Konstruktor wird immer dann automatisch aufgerufen, wenn eine neue Instanz der Klasse „Fahrzeug“ angelegt wird. Alle im Konstruktor enthaltenen Befehle werden dann für diese neue Instanz der Klasse ausgeführt.
- Ein Konstruktor besitzt **denselben** Namen wie die zugehörige Klasse.
- Ein Konstruktor besitzt **keinen** Rückgabewert (auch nicht **void**).

### 2. Destruktor

Prototyp: `~Fahrzeug();`

Implementation:

```
Fahrzeug::~~Fahrzeug()
{
    <Befehle, z.B. zur Freigabe dynamisch
    angeforderten Speichers>
}
```

- Der Destruktor wird immer dann automatisch aufgerufen, wenn eine Instanz der Klasse „Fahrzeug“ nicht mehr gültig ist. Dies ist spätestens beim Programmende der Fall. Alle im Destruktor enthaltenen Befehle werden dann für diese Instanz ausgeführt.
- Ein Destruktor besitzt **denselben** Namen wie die zugehörige Klasse. Dem Namen wird zusätzlich noch eine Tilde (~) vorangestellt.
- Ein Destruktor besitzt **keinen** Rückgabewert (auch nicht **void**).