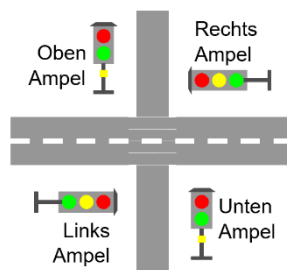


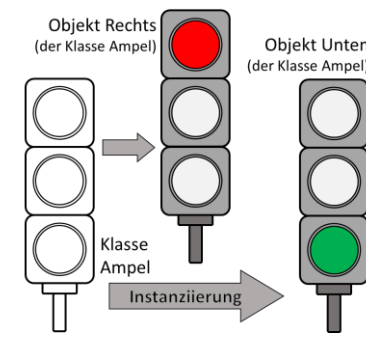
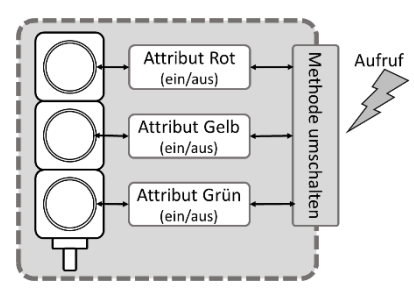
Die Grundidee des OO-Programmierparadigma besteht darin die zu verarbeitenden Daten und die Algorithmen (Methoden) mit denen die Daten verarbeitet werden zu handlichen, wartbaren Einheiten zusammenzufassen.

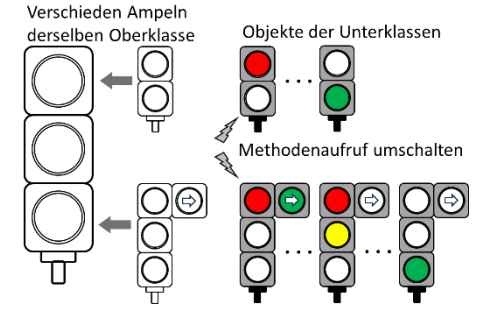
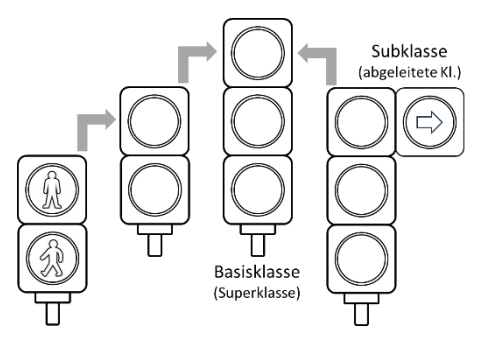
Beispiel Ampelkreuzung	
<p>Problem: Um die Steuerung für einen einfachen Fußgängerüberweg zu programmieren, benötigt man 4 Ampeln (2 Fußgänger und 2 Auto Ampeln). Für die Umsetzung in eine prozedurale Programmiersprache müssten für jede Ampelfarbe ein Speicher deklariert werden (10 Variablen) und somit etwa 14 Funktionen/Prozeduren, die diese Ampeln umschalten. Weiterhin muss für jede Fußgängerampel die Möglichkeit einer Signalanforderung programmiert werden. Alle Signale müssen untereinander synchronisiert werden.</p>	
<p>Lösungsansatz: Bei einer Verkehrsampel bildet der Zustand der Signalleuchten die Daten. (Rot an/aus, Grün an/aus ...) und können durch zwei boolesche Variablen repräsentiert werden. Die dazugehörigen Algorithmen könnten „Ampel-einschalten“ oder „Zustand-umschalten“ oder „Grün-Anfordern“ sein. Diese Daten (auch Attribute) und Operationen (auch Methoden) werden zu einer Einheit zusammengefasst. Von der so entstandenen Klasse werden mehrere konkrete Objekte instanziiert.</p>	

Objektorientierte Programmierung fasst man mit den folgenden wesentlichen Konzepten zusammen:

- ① Klassenbildung (Abstraktion)
- ② Datenkapselung (Information Hiding)
- ③ Vielgestaltigkeit (Polymorphie)
- ④ Vererbung (Spezialisierung)

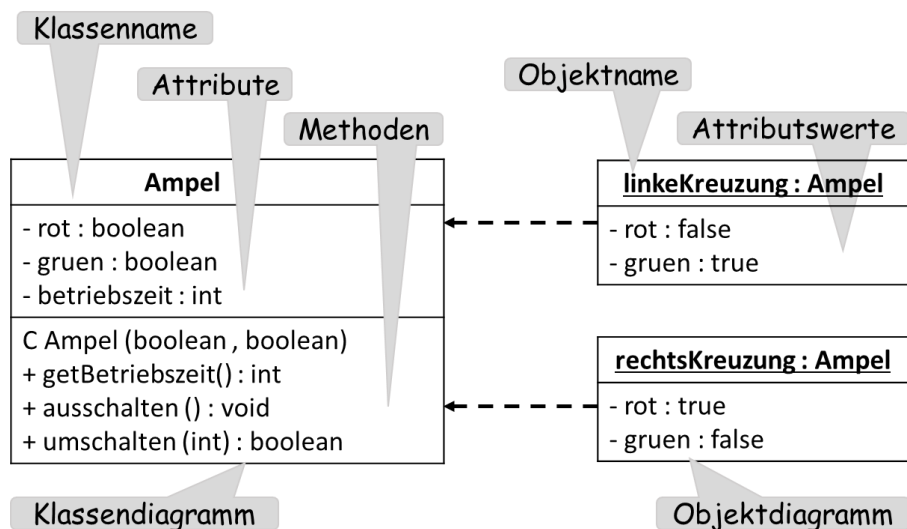
Die nachfolgend am Beispiel der Ampel (-Kreuzung) näher betrachtet werden:

<p>① Klassenbildung (Abstraktion)</p> <p>Bei der Klassenbildung in der objektorientierten Programmierung handelt es sich um das Konzept der Abstraktion, bei dem ähnliche Objekte-Eigenschaften zu einer Klasse zusammengefasst werden. Eine Klasse dient als Bauplan oder Vorlage, um Objekte mit gemeinsamen Merkmalen und Verhalten zu erzeugen.</p> <p>Klasse -> ist der ist eine Beschreibung (Bauplan) für eine Menge gleichartiger Objekte.</p> <p>Objekt -> ist eine konkrete Instanz einer Klasse mit konkreten Attributen.</p>	<p>Beispiel Ampel</p> 
<p>② Datenkapselung (Information Hiding)</p> <p>Datenkapselung ist die Idee des Verbergens von Informationen und der Bereitstellung von Schnittstellen (z.B. Methoden) zur Interaktion mit den Daten umfasst. Durch Datenkapselung werden Daten und Methoden in einer Klasse zusammengefasst und nur ausgewählte Informationen nach außen hin sichtbar gemacht. Dies ermöglicht den Schutz der Daten vor unerlaubtem Zugriff und stellt sicher, dass die Manipulation der Daten nur über definierte Methoden erfolgt.</p>	<p>Beispiel Ampel</p> 

<p>③ Vielgestaltigkeit (Polymorphie)</p>	<p>Beispiel Ampel</p>
<p>Polymorphie bezieht sich in der OOP auf die Fähigkeit von Objekten, sich unterschiedlich zu verhalten, je nachdem, welcher Typ von Objekt sie sind. Polymorphie ermöglicht es, dass verschiedene Klassen mit gemeinsamen Schnittstellen verschiedene Implementierungen für ihre Methoden bereitstellen können. Dadurch können Objekte einer übergeordneten Klasse in unterschiedlichen Kontexten verwendet werden, ohne dass der Programmierer spezifische Details der konkreten Implementierung kennen muss.</p>	
<p>④ Vererbung (Spezialisierung)</p>	<p>Beispiel Ampel</p>
<p>Vererbung ist ein grundlegendes Konzept, bei dem eine Klasse (die sogenannte Subklasse oder abgeleitete Klasse) Eigenschaften und Methoden von einer anderen Klasse (die sogenannte Basisklasse oder Superklasse) erbt. Die Vererbung ermöglicht es, bereits vorhandenen Code wiederzuverwenden und die Struktur und das Verhalten von Klassen hierarchisch zu organisieren. Die abgeleitete Klasse erbt die Merkmale der Basisklasse und kann sie erweitern, überschreiben oder spezialisieren.</p>	

Beim OO-Entwurf und zu Dokumentationszwecken werden meist UML-Diagramme verwendet.

Die **Unified Modeling Language, kurz UML**, ist eine grafische Modellierungssprache zur Spezifikation, Konstruktion, Dokumentation und Visualisierung von Software-Teilen und anderen Systemen. Sie ... ist von der ISO genormt. Die UML besteht aus verschiedenen



Diagrammarten (z.B.: Anwendungsfalldiagramm, Aktivitätsdiagramm, Paketdiagramm, Sequenzdiagramm, Zustandsdiagramm, ...).

Wikipedia (2023): UML, [online] https://de.wikipedia.org/wiki/Unified_Modeling_Language [15.07.2023]

