

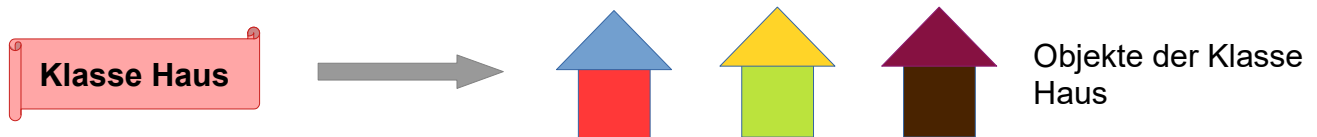
# Einführung und Wiederholung

**Java** ist eine sogenannte objektorientierte Programmiersprache. Das bedeutet, dass der Aufbau und das Zusammenspiel der Bestandteile (**Klassen**, **Objekte**, **Attribute**, **Methoden**,...) der Programmiersprache dem dir bereits bekannten Muster folgt.

Für die Programmierung wird eine Entwicklungsumgebung benötigt, in der Programmiercode geschrieben, ausgeführt und getestet werden kann. Wir werden für den Einstieg **BlueJ** verwenden:

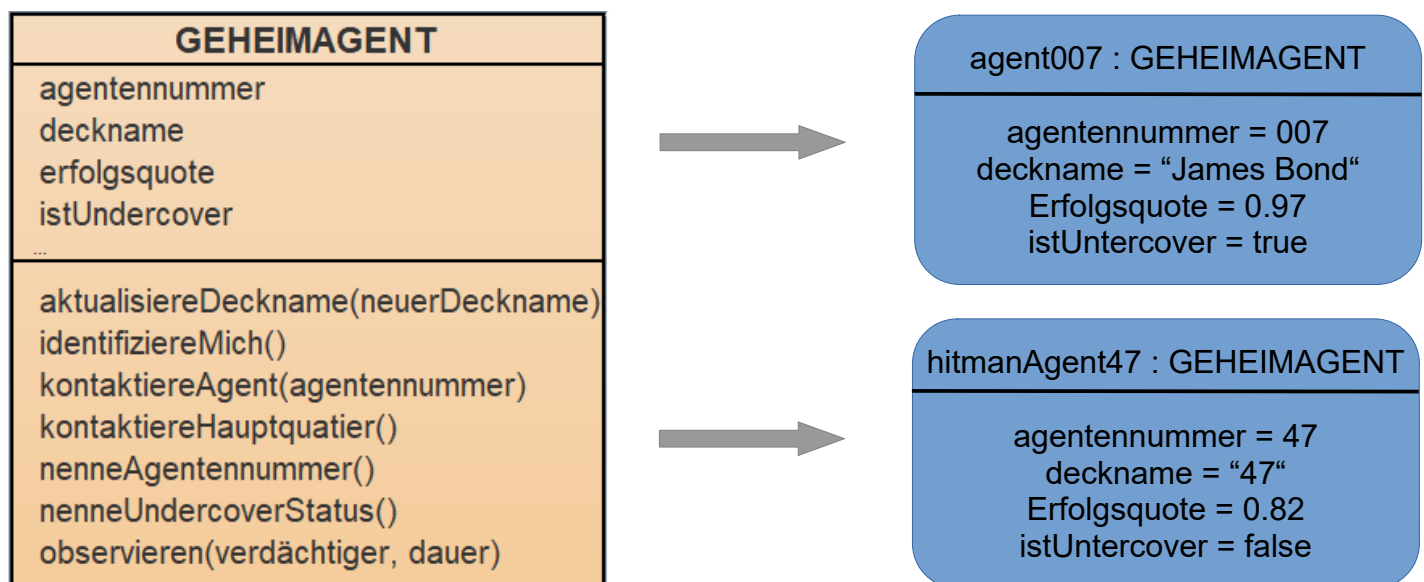
Download für alle Betriebssysteme: <https://www.bluej.org/>

## Wiederholung der Grundlagen

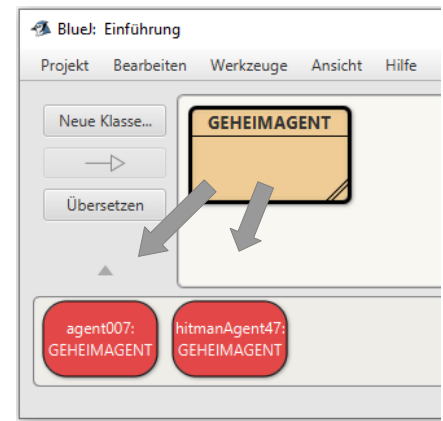


Eine **Klasse** ist ein Bauplan. Sie legt fest, welche **Attribute** (= **Eigenschaften**) und welche **Methoden** (= **Fähigkeiten**) **Objekte** dieser Klasse haben sollen.

Der Aufbau einer Klasse wird durch eine **Klassenkarte** veranschaulicht. Mittels der Klasse kann man (mehrere) Objekte dieser Klasse erzeugen. Sie können sich in ihren **Attributwerten** unterscheiden. Objekte werden durch eine **Objektkarte** veranschaulicht:



Das Programmieren in BlueJ funktioniert nach dem gleichen Schema. Man schreibt/programmiert sich eine Klasse und kann dann mittels dieser Klasse Objekte dieses Klassentyps erstellen.



## Datentypen in Java

Datentypen legen fest, von welcher Art (logischen Struktur) ein Attribute bzw. Variable ist.

| Datentyp       | Bedeutung         | Beispielwerte              | Anmerkung                                                               |
|----------------|-------------------|----------------------------|-------------------------------------------------------------------------|
| <b>int</b>     | Ganze Zahl        | 7 -8                       |                                                                         |
| <b>double</b>  | Dezimalzahl       | 13.24 -0.5                 | Mit Dezimalpunkt                                                        |
| <b>String</b>  | Zeichenkette      | "blau" "Hallo"<br>"5+7" "" | In Anführungszeichen<br><b>Ausnahme:</b> Datentyp wird großgeschrieben! |
| <b>boolean</b> | Wahrheitswert     | true false                 | Eigennamen für diese Werte                                              |
| <b>char</b>    | Einzelnes Zeichen | 'c' '8' '+' ''             | In Hochkommata                                                          |

» weitergehende Informationen zu Datentypen (von Variablen) im Skript zu Variablen

In der Klassenkarte werden die **Datentypen** (analog zum Schema/Klassenkarten von Tabellen bei Datenbanken) mit dem Doppelpunkt : hinter den Namen des Attributs geschrieben.

Dies wird bei Übergabeparametern sowie bei der Rückgabe von Werten bei Methoden analog gehandhabt. Der Rückgabebetyp **void** bedeutet, dass kein Wert von der Methode zurückgegeben wird.

| GEHEIMAGENT                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| agentennummer : int<br>deckname : String<br>erfolgsquote : double<br>istUndercover : boolean<br>...                                                                                                                                                                                                   |
| aktualisiereDeckname(neuerDeckname : String) : void<br>identifiziereMich() : void<br>kontaktiereAgent(agentennummer : int) : void<br>kontaktiereHauptquartier() : void<br>nenneAgentennummer() : int<br>nenneUndercoverStatus() : boolean<br>observieren(verdächtiger : String, dauer : int) : String |

## Methodenaufruf durch Punktnotation

Ein **Methodenaufruf** erfolgt durch die **Punktnotation**. Das bedeutet, dass man ein Objekt dazu veranlasst seine **Fähigkeit auszuführen**, indem man es mit der Punktnotation anspricht:

**agent007.kontaktiereHauptquartier();**

**agent007.observieren("Agent47",10);**

Werte für die  
Übergabeparameter, damit  
agent007 weiß, wen er wie  
lange observieren soll

Bei Methoden mit **Übergabeparameter** muss der Wert für diesen (dem Datentyp entsprechend) in den Klammern übergeben werden.

» weitergehende Informationen zu Übergabeparametern im Skript zu Variablen

» weitergehende Informationen zu Methoden im Skript zu Methoden