**Vererbung**

Dieses Konzept der OOP beschreibt, dass eine Klasse von einer weiteren Klasse erbt.  
Dadurch hat nun die Subklasse die Möglichkeit auf alle Methoden der Vater Klasse zuzugreifen und sie zu verwenden. Dadurch wird redundanter Code vermieden, weil in der Oberklasse die Methoden bereits implementiert wurden. Man versucht die Oberklasse so allgemein wie möglich zu halten, damit auch viele andere Klasse von ihm erben können. Spezielle Fälle werden dann innerhalb der Unterklasse gesondert implementiert.

In meinem Programm erben die „Human“ und die „Saiyajin“ Klasse von der „Fighter“ Klasse, da sowohl „Human“ als auch „Saiyajin“ gleichzeitig Fighter sind. Sowohl „Human“ als auch „Saiyajin“ haben gewisse Spezialattacken, die sie ausführen können. Daher muss diese Methode gesondert innerhalb der jeweiligen Klassen implementiert werden.

**Polymorphie**

Dieses Konzept der OOP baut auf der Vererbung auf. Es besagt, dass Oberklassen auch den Datentypen von Unterklassen annehmen können.

In meinem Programm ist die Oberklasse Fighter, welche alle Attribute und Methoden von allen Kämpfern beinhaltet. Aber nicht jeder Fighter ist gleichzeitig auch ein „Saiyajin“. Ein „Fighter“ kann ein „Saiyajin“ aber auch ein „Human“ sein.

Alle „Saiyajin“ oder aber auch alle „Humans“ sind aber „Fighter“ sodass folgender Ausdruck Sinn ergibt.



Der Nutzen hierin liegt darin, dass man einen „Fighter“ deklarieren kann und zur Laufzeit entsprechend der Eingabe vom User einen „Saiyajin“ oder aber auch einen „Human“ erzeugen kann.

Dadurch ist es möglich alle Methoden vom „Fighter“ auszuführen. Weiterhin ist es möglich überschriebene Methoden innerhalb der Unterklasse auszuführen. In meinem Programm würde der „Human“ bei Ausführung der Methode „makeSpecialAttack“ etwas anderes machen als der „Saiyajin“.