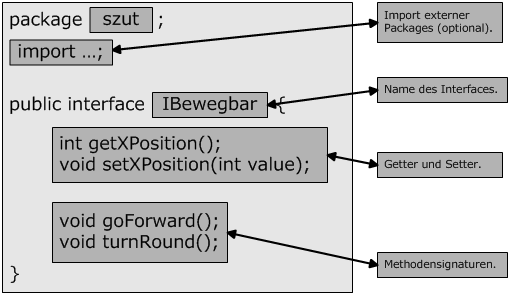
**Interfaces**

Ein Interface beschreibt wie eine Klasse auch einen vom Programmierer selbst erstellten Datentypen. Interfaces sind mit rein abstrakten Klassen zu vergleichen, d.h. es gibt nur statische Attribute oder Konstanten (mit *final* deklarierte Attribute) und keine ausprogrammierten Methoden.

Erbt eine Klasse von zwei anderen Klassen, so spricht man von Mehrfach­vererbung. Mehrfachvererbung kann den Compiler z.B. bei Methoden mit gleicher Signatur in beiden Basisklassen vor unlösbare Aufgaben stellen: Welche der beiden Methoden gilt nun für die erbende Klasse? Dieses Problem könnte man allerdings noch durch ein Überschreiben der Methode lösen. Unlösbar aber ist das Problem bei der Mehrfachvererbung eines gleichnamigen Attributs.

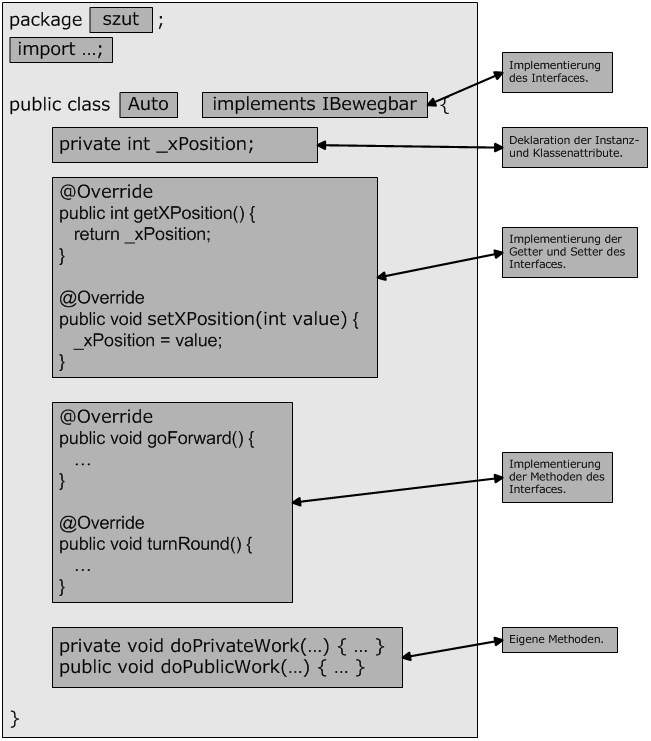
In moderneren Sprachen, zu denen auch Java noch zählt, gibt es keine Mehrfach­vererbung mehr. Statt Mehrfachvererbung gibt es Interfaces bzw. Schnittstellen. Dabei beschreibt ein Interface lediglich, welche Methoden eine Klasse, die es implementiert, auf jeden Fall bereitstellen muss. Eine ein Interface imple­mentierende Klasse geht somit einen Vertrag zur Realisierung der im Interface festgelegten Methoden ein.

Alle Methoden eines Interfaces sind öffentlich. Aus diesem Grunde wird der Zugriffsspezifizierer weggelassen. Bei den Methoden werden nur die Methodensignaturen in einem Interface aufgeführt. Diese werden wie Anweisungen mit einem Semikolon abgeschlossen.



Eine Klasse kann zwar nur von einer anderen Klasse erben, aber mehrere Interfaces implementieren. Dabei ist es auch kein Problem, wenn beide Interfaces eine Methode mit gleicher Signatur beinhalten, denn die Klasse selbst übernimmt die Implementierung. Es ist sogar möglich, dass auch die Basisklasse diese gleichsignierte Methode enthält. Dann ist sie bereits über die Basisklasse in der Klasse implementiert.

Bei der Implementierung der Methoden aus einem Interface ist zu beachten, dass alle Methoden aus dem Interface mit dem Zusatz *@Override* zu versehen sind. Dies kennzeichnet die Methode für den Compiler als eine Methode, die eine in der Vererbungshierarchie höher stehende Methode (nämlich die aus dem Interface) überschreibt.



Interfaces können zwar erben und vererben, sie können allerdings nicht instanziiert werden. Aber das ist eigentlich auch klar, denn sie besitzen keine ausprogrammierten Methoden, sondern nur die Signaturen.