4.3 Delegations-Eventmodell

Sobald ein Programm nicht mehr sequentiell abläuft, da es auf Ereignisse, wie Tastatur- oder Mauseingaben, wartet, wird es als ereignisgesteuertes Programm bezeichnet. Ereignisgesteuerte Programme haben Methoden, die inaktiv sind, bis ein bestimmtes Event sie aktiviert. Solche Methoden nennt man Callbackmethoden. Ein Programm, welches Callbackmethoden verwenden will, muss einen Eventlistener implementieren, wodurch beim Auftreten des erwarteten Events die zugehörige Callbackmethode aufrufen werden kann. Die Callbackmethode wird in der Klasse, welche den Eventlistener implementiert deklariert, wodurch selbst bestimmt werden kann, was passieren soll, sobald die Callbackmethode aufgerufen wird. Im Tichuprogramm wird zum Beispiel zum Markieren der Karten der PointerDownListener benötigt, da dieser bei einem Mausklick ein Event auslöst, über das auch die Position des Mausklicks abgefragt werden kann. In diesem Beispiel implementiert die Klasse Board, die für die Grafische Oberfläche zuständig ist, den PointerDownListener, weshalb sie die Methode onPointerDown() haben muss. Danach muss der Klasse View, welche dem Fenster, in dem die Grafik gezeichnet wird, entspricht, ein PointerDownListener hinzugefügt werden, welcher in diesem Fall das Objekt der Klasse Board ist, was nur möglich ist, weil die Klasse Board den EventListener implementiert. Durch das hinzufügen des Board Objekts als PointerDownListener zur Klasse View, kann die Klasse View nun die Callbackmethode onPointerDown() in der Board-Klasse aufrufen, falls mit der Maus in das Fenster geklickt wird.

Als Eventmodell bezeichnet man das Verfahren, mit welchem dieses Konzept mit Callbackmethoden programmiertechnisch umgesetzt wird. Da das klassische Eventmodell nicht objektorientiert ist und daher nicht wirklich in eine Klassenstruktur passt, entwickelten die Java-Entwickler in der Java Version 1.1 das Delegations-Eventmodell, welches viel besser zum objektorientierten Programmieren passte.