

7 Tarantel, du musst dich entscheiden — bedingte Anweisungen

Konzepte in diesem Kapitel

Algorithmik/Java: einseitige und zweiseitige bedingte Anweisung

Im Alltag entscheidet oft die konkrete Situation über unser Tun. Die Entscheidung darüber, ob wir eine Handlung ausführen oder nicht, hängt von bestimmten Bedingungen ab, z.B.:

- **Wenn** es regnet, **dann** nehme ich den Regenschirm mit.
- **Wenn** ich kein Brot mehr habe, **dann** kaufe ich ein neues.
- **Wenn** die Mauer nicht höher als zwei Meter ist, **dann** springe ich.
- **Wenn** die Ampel rot zeigt, **dann** bleibe ich stehen, **sonst** gehe ich weiter.
- **Wenn** Schnee liegt, **dann** fahre ich mit der Bahn, **sonst** nehme ich das Fahrrad.

Auch eine Tarantel muss häufig Entscheidungen treffen. Das ermöglichen ihr *bedingte Anweisungen*.

7.1 Tu ich's oder tu ich's nicht — die einseitige bedingte Anweisung

Die Tarantel will einen Schritt gehen. Damit sie es auch überlebt, muss sie sicher sein, dass keine Klatsche vor ihr lauert:

umgangssprachlich	Java	allgemeine Java-Syntax
wenn vorne keine Klatsche lauert geh einen Schritt weiter	<pre>if(!klatscheVorn()) { schritt(); }</pre>	<pre>if(<i>Bedingung</i>) { <i>Anweisungsblock</i> }</pre>

Sie geht nur dann einen Schritt, wenn keine Klatsche vor ihr lauert. Ansonsten wird der Befehl einfach übersprungen.

Aufgabe 1: Tragen Sie die allgemeine Java-Syntax der einseitigen bedingten Anweisung in Ihr Arbeitsblatt ein. Formulieren und testen Sie dann folgende Anweisungen:

- Wenn rechts ein Hindernis liegt, drehe dich nach links.
- Wenn am Platz eine Fliege liegt, nimmt sie auf und drehe dich um 180°.
- Wenn vorne eine Klatsche lauert, spucke eine Fliege aus und rücke ein Feld nach rechts.

Aufgabe 2: Erklären Sie den Unterschied zwischen einer bedingten Anweisung und einer Schleife.

7.2 Die Qual der Wahl — die zweiseitige bedingte Anweisung

Wieder will die Tarantel einen Schritt gehen. Diesmal soll sie nicht einfach stehen bleiben, wenn sie vor einer Klatsche steht, sondern diese umgehen. Sie steht also vor der Entscheidung:

umgangssprachlich	Java	Java-Syntax allgemein
wenn vorne eine Klatsche lauert umgehe die Klatsche	<code>if(klatscheVorn()) { klatscheUmgehen(); }</code>	<code>if(<i>Bedingung</i>) { <i>Anweisungsblock 1</i> }</code>
sonst geh einen Schritt weiter	<code>else { schritt(); }</code>	<code>else { <i>Anweisungsblock 2</i> }</code>

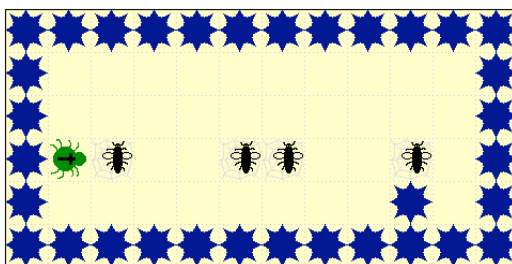
Die Tarantel muss also entweder die eine oder die andere Handlung ausführen. Ein Überspringen der gesamten Anweisung ist nicht möglich.

Aufgabe 3: Tragen Sie die allgemeine Java-Syntax der zweiseitigen bedingten Anweisung in Ihr Arbeitsblatt ein. Formulieren und testen Sie dann folgende Anweisungen:

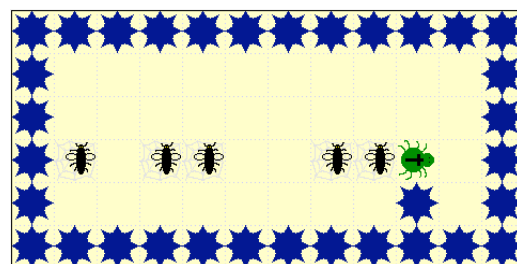
- Wenn vorne eine Klatsche lauert, wende, sonst spucke eine Fliege aus.
- Wenn rechts ein Hindernis liegt, drehe dich nach links, sonst drehe dich nach rechts.
- Wenn am Platz eine Fliege liegt, nimm sie auf und gehe einen Schritt, sonst spucke eine aus und wende.

Aufgabe 4: Schreiben Sie die Methode `fliegeSuchen2()`, mit der die Tarantel auf Fliegen-suche alle eventuellen Hindernisse umgeht.

Aufgabe 5: Die Tarantel hat Langeweile und beschließt das Fliegenmuster auf ihrem Weg zu tauschen. Schreiben Sie dafür zuerst die Methode `fliegeTauschen()` für einen einzelnen Platz:



Anfang



Ende

Aufgabe 6 (Stammgruppe): Programmieren Sie die gesamte Lösung.

Aufgabe 7 (Stammgruppe): Testen Sie das Programm mit verschiedenen Fliegenmustern. Natürlich müssen Sie die Tarantel mit ausreichend Fliegen auf den Weg schicken. Suchen Sie dazu in der Schnittstelle der Klasse `Spinne` eine geeignete Methode.