

Verknüpfung von Bedingungen

In Java können mehrere Bedingungen in einer bedingten Anweisung/(mehrfachen) Fallunterscheidung miteinander verknüpft werden.

```
if ( zeugnis.wenigerAlsZweiFünfen() && zeugnis.keineSechs() ) {          if ( x > 5 && x <10 ) {...}
    schüler.klasseBestanden() }
```

Mithilfe eines ANDs (in Java: **&&**) bewirkt, dass beide Bedingungen erfüllt sein müssen, sodass der Code ausgeführt wird.

```
if ( oberstufe.wähleMusik() || oberstufe.wähleKunst() ) {
    }
```

Mithilfe eines ODERs (in Java: **||**) bewirkt, dass mindestens eine Bedingungen erfüllt sein muss, sodass der Code ausgeführt wird.

```
if ( ! person.unter16() ) {          if ( farbe != "rot" ) {
    bier.trinken() }                farbe = "rot" }
```

Mithilfe eines Ausrufezeichens **!** kann in Java eine Bedingung „umgekehrt“ werden.

Legende für die nachfolgenden Aufgaben:

- **Klassen**
- **Attribute**
- **Methoden und Übergabeparameter**
- zu verwendende Werte
- **Tipps und Hinweise**
- **Java-Code**

Aufgabe 1:

In dieser Aufgabe soll der Feuerball nur sichtbar werden, wenn der Controller oder die Leertaste gedrückt gehalten wird. Anschließend sollen die Gegner (turtle, plant, bomb) durch den Feuerball vernichtbar gemacht werden.

Gehe in die Klasse **SPIELFIGUR** und schreibe zwei eigene Methoden **feuer()** und **feuerEnde()**, in denen du den **feuerball** sichtbar und unsichtbar setzt.

Wechsel in den Klasse **SUPERMARIO** zur Methode **tick()**. Implementiere eine Fallunterscheidung in einer Fallunterscheidung nach folgendem Schema: Falls der Joystick aktiviert ist, soll überprüft

werden ob der Switch gedrückt ist. Falls ja, rufe die Methode **feuer()** auf; falls nicht rufe die Methode **feuerEnde()** auf. Falls der Joystick nicht aktiviert ist, soll überprüft werden ob die Leertaste (Tastenummer: 30) gedrückt ist **(suche hierfür nach einer passenden Methode aus der Klasse SPIEL)**. Falls ja, rufe die Methode **feuer()** auf; falls nicht rufe die Methode **feuerEnde()** auf.

Wechsel in die Klasse **SPIELFIGUR**. Setze den **feuerball** im Konstruktor direkt nach dem Erzeugen auf unsichtbar. Verändere die Sichtbarkeit des Attributs **feuerball** von **private** auf **public**.

Wechsel zurück in die Klasse **SUPERMARIO**. Dekлариere ein ganzzahliges Attribut **leben** und initialisiere es mit 0. Gehe zur Methode **tick()**. Iteriere über alle Bomben und überprüfe, ob **mario** eine der Bomben berührt. Falls ja, erhöhe **leben** um 1 und setze **mario** wieder an seinen Anfangspunkt. Iteriere über alle Pflanzen und überprüfe, ob **mario** eine der Pflanzen berührt. Falls ja, erhöhe **leben** um 1 und setze **mario** wieder an seinen Anfangspunkt. Überprüfe weiterhin in dieser Schleife, ob der **feuerball** (auf den jetzt öffentlichen (public) Feuerball kann mittels **mario.feuerball** zugegriffen werden) sichtbar ist UND dieser eine der Pflanzen berührt. Falls ja, setze die Pflanze außerhalb des Spielfelds. Iteriere über alle Schildkröten und überprüfe, ob der **feuerball** sichtbar ist UND eine der Schildkröten berührt ODER der **mario** auf einer der Schildkröten steht **(suche hierfür nach einer passenden Methode aus der Klasse SPIEL)**. Setze dann diese Schildkröte außerhalb des Spielfelds.

Starte das Spiel mit und ohne Joystick und überprüfe, ob der Feuerball entsprechend durch Drücken der Leertaste oder des Joysticks auftaucht. Überprüfe weiterhin, ob der Mario beim Berühren von Pflanzen und Bomben stirbt; ob der Feuerball die Schildkröten und Pflanzen zerstört; und ob der Mario beim „Draufhüpfen“ die Schildkröten eliminiert.

Aufgabe 2:

In dieser Aufgabe soll das Ende des Spiels definiert werden. Weiterhin soll die Musik in das Spiel eingebaut werden.

Dekлариere zwei Attribute **finish** und **musik** vom Typ **SOUND**. Initialisiere die beiden Attribute mit den passenden Sounddateien. Beim Erzeugen des **SUPERMARIO**-Spiels soll direkt die **musik** abgespielt werden.

Wechsel zur Methode **tick()**. Überprüfe, ob der **mario** die Fahne berührt, indem du überprüfst ob der **mario** den **Punkt** (4070 | 300) beinhaltet **(suche hierfür nach einer passenden Methode aus der Klasse BILD)**. Falls ja, stoppe den Ticker; stoppe die **musik** und spiele den **finish**-Sound ab.

Teste, ob die Musik abgespielt wird. Überprüfe weiterhin, ob das Spiel beim Berühren der Fahne beendet wird (Du kannst für diesen Test den Mario direkt am Ende der Karte starten lassen).

Aufgabe 3:

Kurze Erklärung, was mit dieser Aufgabe realisiert werden soll (Agil: User Story)

Beschreibung des ersten Schritts

Zweiter Schritt

Dritter Schritt

Beschreibung eines Test passend zu der Aufgabe, sodass der Schüler selbstständig überprüfen kann, ob die Aufgabe richtig erledigt wurde.

Aufgabe 4:

Kurze Erklärung, was mit dieser Aufgabe realisiert werden soll (Agil: User Story)

Beschreibung des ersten Schritts

Zweiter Schritt

Dritter Schritt

Beschreibung eines Test passend zu der Aufgabe, sodass der Schüler selbstständigüberprüfen kann, ob die Aufgabe richtig erledigt wurde.

Aufgabe 5:

Kurze Erklärung, was mit dieser Aufgabe realisiert werden soll (Agil: User Story)

Beschreibung des ersten Schritts

Zweiter Schritt

Dritter Schritt

Beschreibung eines Test passend zu der Aufgabe, sodass der Schüler selbstständigüberprüfen kann, ob die Aufgabe richtig erledigt wurde.

Aufgabe 6:

Kurze Erklärung, was mit dieser Aufgabe realisiert werden soll (Agil: User Story)

Beschreibung des ersten Schritts

Zweiter Schritt

Dritter Schritt

Beschreibung eines Test passend zu der Aufgabe, sodass der Schüler selbstständig überprüfen kann, ob die Aufgabe richtig erledigt wurde.

Aufgabe 7:

Kurze Erklärung, was mit dieser Aufgabe realisiert werden soll (Agil: User Story)

Beschreibung des ersten Schritts

Zweiter Schritt

Dritter Schritt

Beschreibung eines Test passend zu der Aufgabe, sodass der Schüler selbstständig überprüfen kann, ob die Aufgabe richtig erledigt wurde.

Aufgabe 8:

Kurze Erklärung, was mit dieser Aufgabe realisiert werden soll (Agil: User Story)

Beschreibung des ersten Schritts

Zweiter Schritt

Dritter Schritt

Beschreibung eines Test passend zu der Aufgabe, sodass der Schüler selbstständigüberprüfen kann, ob die Aufgabe richtig erledigt wurde.

Aufgabe 9:

Kurze Erklärung, was mit dieser Aufgabe realisiert werden soll (Agil: User Story)

Beschreibung des ersten Schritts

Zweiter Schritt

Dritter Schritt

Beschreibung eines Test passend zu der Aufgabe, sodass der Schüler selbstständigüberprüfen kann, ob die Aufgabe richtig erledigt wurde.

Aufgabe 10:

Kurze Erklärung, was mit dieser Aufgabe realisiert werden soll (Agil: User Story)

Beschreibung des ersten Schritts

Zweiter Schritt

Dritter Schritt

Beschreibung eines Test passend zu der Aufgabe, sodass der Schüler selbstständig überprüfen kann, ob die Aufgabe richtig erledigt wurde.

Zusatzaufgabe 1:

Kurze Erklärung, was mit dieser Aufgabe realisiert werden soll (Agil: User Story)

Beschreibung des ersten Schritts

Zweiter Schritt

Dritter Schritt

Beschreibung eines Test passend zu der Aufgabe, sodass der Schüler selbstständigüberprüfen kann, ob die Aufgabe richtig erledigt wurde.

Für ganz schnelle SchülerInnen:

Werdet kreativ!

Mögliche Ideen: