



DENKEN VERSTEHEN LERNEN

Computational Thinking in der Grundschule

Grundschulpraktikum (B.Ed. und B.Sc.)

17.11.2016



Einheit C5

Verzweigungen (Unplugged)



Übersicht

- Wir wissen nicht immer im Voraus, wie manche Dinge sein werden, wenn wir unsere Programme ablaufen lassen.
- Dafür werden Verzweigungen eingesetzt.
- In dieser Einheit wird gezeigt, wie Verzweigungen verwendet werden können, um ein Programm auf spezifische Informationen zuzuschneiden.



Lernziele

Allgemeine Kompetenzen:

- **Definieren**, unter welchen **Umständen** bestimmte **Teile eines Programms laufen** sollten und unter welchen Umständen diese nicht laufen sollten.
- Basierend auf bestimmten Kriterien **entscheiden**, ob eine **Bedingung erfüllt** ist.
- Ein **Programm durchgehen** und das **Ergebnis** für eine gegebene Eingabe **bestimmen**.



Einteilung der Unterrichtsstunde

1. Einstieg (15 Min.)
 - A. Wiederholung
 - B. Neue Wörter
 - C. Unter einer Bedingung
2. Gruppenarbeit: Verzweigungen (30 Min.)
3. Abschluss (5 Min.)
 - A. Kurzgespräch: Was haben wir gelernt?
 - B. Wiederholung: Neue Wörter
4. Test (5 Min.)



Wiederholung

Rekapitulieren Sie mit der Klasse die letzte Stunde:

- Stellen Sie Fragen und lassen Sie die SchülerInnen diese in kleinen Gruppen diskutieren:
 - Was haben wir letztes Mal gemacht?
 - Was hätten ihr gerne noch gemacht?
 - Sind euch nach der vorherigen Stunde noch Fragen eingefallen?
 - Was hat euch an der vorherigen Stunde am besten gefallen?



Neue Wörter

Schreiben Sie den Begriff und seine Bedeutung an die Tafel.
Sprechen Sie vor und lassen Sie die Klasse wiederholen.

Bedingte Anweisung

be-ding-te An-wei-sung

Anweisung, die nur unter bestimmten Bedingungen ausgeführt wird



Unter einer Bedingung

- Machen Sie der Klasse folgendes Angebot: Wenn sie es schafft für 30 Sek. völlig still zu sein, werden Sie ...
 - A. eine Opernarie singen, oder
 - B. noch weitere 5 Min. warten bevor die Stunde anfängt, oder
 - C. einen Handstand machen.
- Fangen Sie sofort an zu zählen.
- Schaffen es die Schüler so lange still zu sein, sagen Sie Ihnen:
„Ihr wart erfolgreich, deshalb **bekommt** ihr die Belohnung.“
- Ansonsten sagen Sie ihnen, dass sie nicht die ganzen 30 Sek. still waren und deshalb die Belohnung **nicht bekommen**.



Unter einer Bedingung

- Fragen Sie die Klasse: Was war die Bedingung für die Belohnung?
 - Die Bedingung war „**WENN** ihr für 30 Sekunden still seid“.
 - Die englische Bezeichnung für „wenn“ ist „if“, das Wort „if“ findet man in vielen Programmiersprachen.
 - Wenn die Klasse still war, war die **Bedingung wahr**, und sie bekommt die **Belohnung**.
 - Wenn nicht, war die **Bedingung falsch**, und es gibt **keine Belohnung**.
- Fallen der Klasse noch andere Bedingungen ein? Beispiele:
 - „Wenn du mein Alter richtig errätst, applaudiert dir die Klasse.“
 - „Wenn ich die Antwort weiß, kann ich meine Hand heben.“



Unter einer Bedingung

- Erläutern Sie: Manchmal möchten wir auch angeben, was passiert wenn die „if“-Bedingung nicht wahr ist.
 - Diese zusätzliche Anweisung wird „else“-Anweisung genannt.
 - „else“: Englisch für „sonst“
 - Trifft die „if“-Bedingung nicht zu, schauen wir beim „else“ nach, was zu tun ist.
 - Beispiel:
 - „**WENN** (if) ich eine 7 ziehe, klatscht jeder, und **SONST** (else) sagt jeder „Ooohhh.““
 - Probieren Sie es aus (Karte ziehen und auf Reaktion warten)



Unter einer Bedingung

- Die Klasse soll analysieren, was gerade passiert ist:
 - Was war das „if“?
 - Was das „else“?
 - Wurde die „if“-Bedingung erfüllt? Welche Anweisung wurde ausgeführt (die bei „if“ oder die bei „else“)?
- Erläutern Sie, dass es noch eine weitere Möglichkeit gibt:
 - Was wenn ich will, dass ihr klatscht, wenn ich eine 7 ziehe, und sonst, wenn ich etwas kleiner als 7 ziehe, ruft ihr „JUHU!“, und sonst sagt ihr „Ooohhh“?
 - Dafür haben wir die Ausdrücke „if“ („wenn“), „else if“ („sonst, wenn“) und „else“ („sonst“).



Unter einer Bedingung

- „if“ ist die erste Bedingung
- „else if“ wird nur betrachtet, wenn die erste Bedingung falsch ist
- „else“ am Ende wird nur betrachtet, wenn nichts davor wahr ist
- solche „if“-„else“-Anweisungen nennen wir:

Verzweigung
Ver-zwei-gung



Gruppenarbeit: Verzweigungen

Einleitung:

- Hier geht es um Algorithmen/Programme, die von Farbe, Wert, oder der Unterscheidung Schwarz/Rot einer gezogenen Spielkarte abhängig sind.
- Abhängig von diesen Eigenschaften einer Karte werden Punkte vergeben oder abgezogen.
- Beispielalgorithmus:

```
Wenn gilt: (KARTE ist ROT)
  Schreibe DEINEM Team 1 Punkt gut
Sonst:
  Schreibe ANDEREM Team 1 Punkt gut
```



Gruppenarbeit: Verzweigungen

Ablauf der Gruppenarbeit:

1. Teilen Sie die Klasse in Teams ein.
2. Geben Sie jedem Team einen Stapel Karteikarten/leere Blätter.
3. Zeigen Sie der Klasse eines Ihrer Programme an der Tafel.

Vorarbeit: Denken Sie sich einige Programme aus.

4. Lassen Sie die Teams reihum Karten ziehen und je nach Vorgabe des Programms den neuen Punktstand aufschreiben.
5. Machen Sie das Gleiche mit anderen Programmen, die sie sich zusammen mit der Klasse überlegen.



Gruppenarbeit: Verzweigungen

Erweiterung:

- Wenn die Klasse etwas Übung mit Verzweigungen hat:
- Probieren Sie mit ihr verschachtelte Verzweigungen aus.
- Beispielalgorithmus:

```
Wenn gilt: (KARTE ist ROT)
  Schreibe DEINEM Team 1 Punkt gut
Sonst:
  Wenn gilt: (KARTE hat Wert höher als 9)
    Schreibe ANDEREM Team 1 Punkt gut
  Sonst:
    Schreibe DEINEM Team die Anzahl Punkte auf
    der Karte gut
```



Gruppenarbeit: Verzweigungen

Variation: Zeigen Sie/Entwickeln Sie mit der Klasse Algorithmen bzw. Programme in folgenden Darstellungen:

- als Algorithmus, der deutscher Sprache ähnelt (s. die Bsp.e oben)
- als Pseudocode:

```

If (card.color == RED) {
    points.yours = points.yours + 1;
}
Else {
    If (card.value > 9) {
        points.others = points.others + 1;
    }
    Else {
        points.yours = points.yours + card.value;
    }
}

```




Gruppenarbeit: Verzweigungen

Variation (Fortsetzung):

- als Programmcode einer echten Programmiersprache (z.B. C):

```
enum Color {
    RED, BLACK
};
struct Card {
    Color color;
    int value;
};
...
Card card = drawCard();
int yourPoints = 0, otherPoints = 0;
if (card.color == RED) {
    yourPoints++;
} else {
    if (card.value > 9) ...
}
```

(Dies nur als kleiner Einblick in die „echte“ Programmierung.)



Kurzgespräch: Was haben wir gelernt?

Diskutieren Sie:

- Wenn ihr das in Blockly programmiert: Was müsst ihr um die Verzweigungen herum hinzufügen, damit der Code mehr als einmal abläuft?
- Was macht ihr im Alltag nur unter bestimmten Bedingungen?
- Wenn ihr etwas tun sollt, wenn der Wert einer Karte größer als 5 ist, und ihr zieht eine 5, ist die Bedingung dann erfüllt?
- Bedingungen sind entweder „wahr“ oder „falsch“: Eine Bedingung kann nicht so bewertet werden, dass „Banane“ herauskommt.



Kurzgespräch: Was haben wir gelernt?

- Wenn es um mehrere Kombinationen von Bedingungen geht, können wir etwas verwenden das verschachtelte Verzweigungen heißt.
 - Was bedeutet das?
 - Könnt ihr ein Beispiel geben, wo wir das während der Übung gesehen haben?
- Welchen Teil der Übung fandet ihr am besten?



Wiederholung: Neue Wörter

- Fragen Sie: Für welche dieser Beschreibungen haben wir heute ein Wort gelernt?
 - „Am Anfang einer Textzeile zusätzlichen Leerraum einfügen“
 - „Eine Kombination aus Gelb und Grün“
 - „Anweisungen, die nur unter bestimmten Bedingungen ausgeführt werden“
- ... und wie lautet dieses Wort?



Zusätzliche Lernangebote

Wahr/Falsch-Fangen:

Vorgehen:

1. Die Schüler stellen sich auf wie beim Spiel „Ochs vorm Berg“: einer als Rufer vor den anderen, diese hinter einer Grundlinie.
2. Der Rufer wählt eine Bedingung. Jeder, der die Bedingung erfüllt, darf einen Schritt vorwärts machen. Beispiele:
 - „Wenn du einen Gürtel trägst, mache einen Schritt vorwärts.“
 - „Wenn du Sandalen trägst, mache einen Schritt.“
3. Die Bedingungen können auch in etwas anderer Form kommen:
 - „Wenn du *nicht* blond bist, schreite vor.“



Zusätzliche Lernangebote

Verschachteln:

Vorgehen:

1. Teilen Sie die SchülerInnen in Paare/kleine Gruppen ein.
2. Sie sollen „wenn“-Anweisungen zum Kartenspielen auf Papierstreifen schreiben, z.B.:
 - „Wenn die Farbe der Karte Pik ist“
 - „Wenn die Karte rot ist“
3. Dann sollen sie Papierstreifen mit möglichen Ergebnissen beschreiben, z.B.:
 - „Füge einen Punkt hinzu“
 - „Ziehe einen Punkt ab“



Zusätzliche Lernangebote

4. Die SchülerInnen wählen drei von jeder Art Papierstreifen und drei Spielkarten aus. Sie merken sich die Reihenfolge, in der sie die Spielkarten gezogen haben.
5. Die SchülerInnen schreiben auf drei Blätter je ein Programm. Dabei verwenden sie nur die Anweisungen auf den ausgewählten Papierstreifen.
Ermuntern Sie sie, auch einige „wenn“-Anweisungen innerhalb anderer „wenn“-Anweisungen zu verwenden.
6. Zum Schluss lassen die SchülerInnen ihre Programme mit den vorher gezogenen Spielkarten ablaufen (die Spielkarten immer in der gleichen Reihenfolge ziehen).



Zusätzliche Lernangebote

7. Lassen Sie sie feststellen:

- Haben irgendwelche zwei Programme dasselbe Ergebnis?
- Haben irgendwelche Programme verschiedene Ergebnisse?



Fragen...?



Danke.

Kontakt:

Julian Jabs

B221

Sand 13, 72076 Tübingen

julian.jabs@uni-tuebingen.de