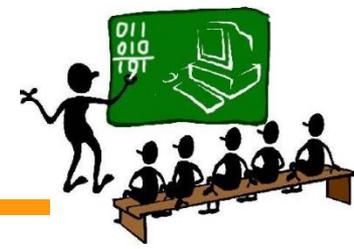
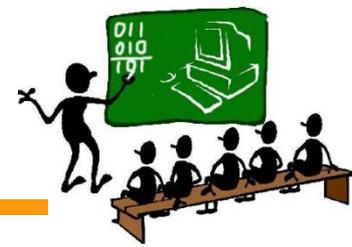


Peter Brichzin

Aufgabenkultur

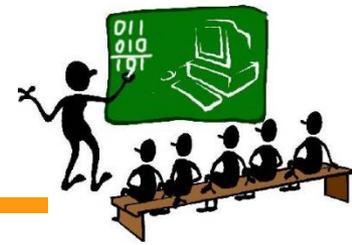


- Erwerb neuer Kompetenzen **Lernaufgaben**
- Einüben neuer Inhalte und ihrer Vernetzung mit Vorangegangenen **Übungsaufgaben**
- Ermittlung des Leistungsstands als wichtige Rückmeldung über den Erfolg bzw. Misserfolg des Lehr- und Lernprozesses **Aufgaben zur Diagnose / zum Leisten**
- Motivation sich mit informatischen Inhalten auseinander zu setzen
- Kollegialer Austausch, Werbung für das Fach
- Binnendifferenzierung, ... **Übergreifende Aspekte**

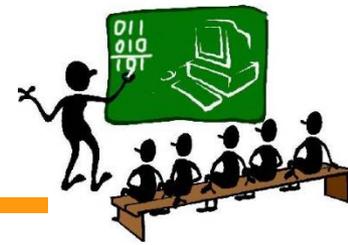


LOGIN Heft 176 / 177 (2013), Berlin 2014

Büchter, A.; Leuders, T.: Mathematikaufgaben selbst entwickeln – Lernen fördern, Leistung überprüfen. Berlin: Cornelsen Scriptor, 2005.



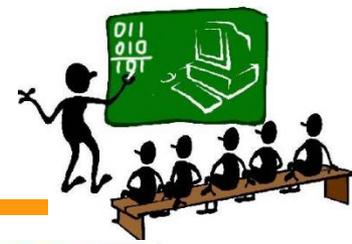
Kontextbezogene Aufgaben – Medizinische
Informatik, Strichcode-Scanner, Zeichenprogramm
von Kerstin Strecker, LOGIN Heft 176 / 177 (2013),
Berlin 2014



Zweidimensionale Felder

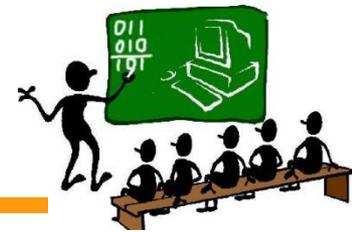
140	140	140	140	140
140	80	255	80	140
140	255	255	255	140
255	80	255	80	255
255	255	80	255	255

115	115	115	115	115
115	175	0	175	115
115	0	0	0	115
0	175	0	175	0
0	0	175	0	0



Zweidimensionale Felder



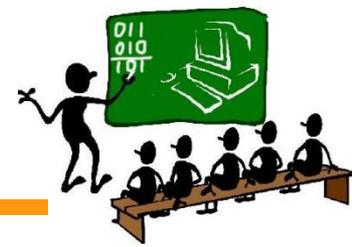


Barcode

Der hier abgebildete EAN-8-Code besteht aus ein von schwarzen und weißen Strichen, wobei jeder fern 0 bis 9 eine Folge von vier Strichbreiten abwe in schwarz und weiß zugeordnet ist (siehe Tabelle). Die längeren, grau unterlegten Striche begrenzen den Code und dienen der Orientierung (drei Striche am Anfang und Ende, fünf in der Mitte).



Ziffer	Strichbreiten	Ziffer	Strichbreiten
0	3211	5	1231
1	2221	6	1114
2	2122	7	1312
3	1411	8	1213
4	1132	9	3112

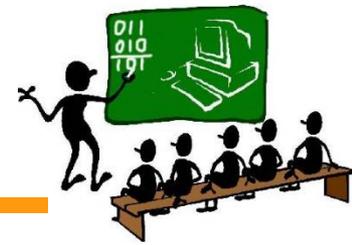


Aufgabe Gruppe 1: „Recherche Strichcode“

Ziel: Aufbau des EAN-8-Strichcodes verstehen und diesen in eine Ziffernfolge übersetzen können.

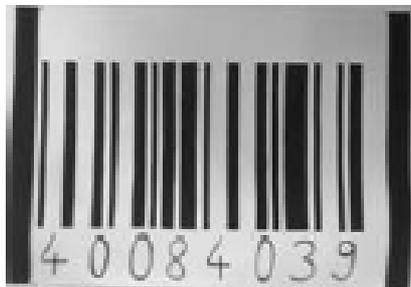
Recherchiert im Netz Informationen über den EAN-8-Code. Versucht mit eurem gewonnenen Wissen die Strichcodes der Produkte am Lehrertisch selbst in die darunter stehende Ziffernfolge zu übersetzen. Bei Fragen stehen euch Hilfekarten zur Verfügung.

Fertigt eine Präsentation an, mit der ihr euer Wissen, wie die schwarz-weißen Strichfolgen in Zahlen übersetzt werden, den anderen erläutern könnt.



Aufgabe Gruppe 2: „Strichcode scannen“

Ziel: Lesen eines Strichcodes mit Lichtsensoren und Speichern des Ergebnisses in einer Tabelle.



Der EAN-8-Strichcode besteht abwechselnd aus weißen und

schwarzen Strichen. Die Breite des Striches bestimmt die Ziffer.

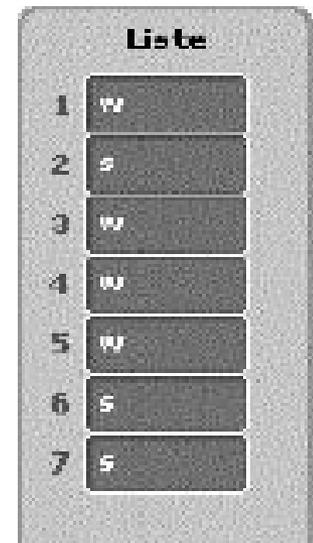
Ein Strich der Breite 1 wird als „w“ (weiß) und ein Strich der Breite 3 als „s“ (schwarz) interpretiert.

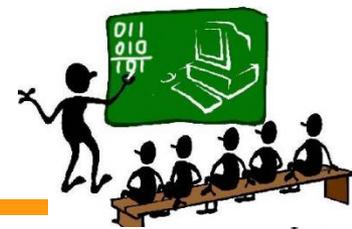
Verwendet den NXT-Roboter mit dem Berührungssensor. Den Roboter stellt man auf dem Lehrertisch. Ein Lichtsensor liest die Breite des Striches ein. Halte die beiden Lichtsensoren auf die ersten beiden Striche und lese die Daten der Sensoren auf Knopfdruck ein. Halte danach die Lichtsensoren auf die nächsten beiden Striche usw.

Wandelt die eingelesenen Daten in die Information „s“ für Schwarz oder „w“ für Weiß um und trage sie in eine Liste ein. Da der Strichcode-Scanner Striche der Breite 1 einliest, entspricht ein Strich der Breite 3 also 3 Einträgen in der Liste.

Beispiel:

Für die Ziffer 4 würde die Liste wie im rechten Bild aussehen:

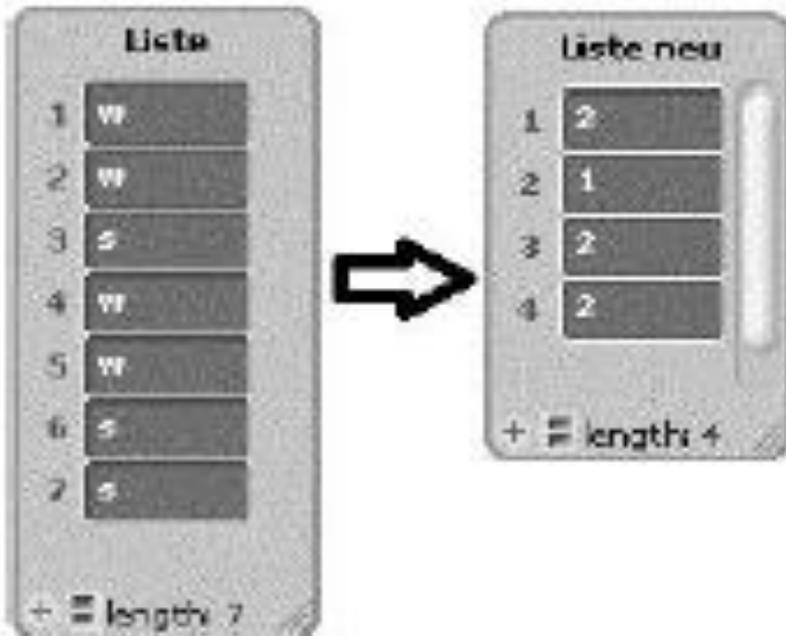




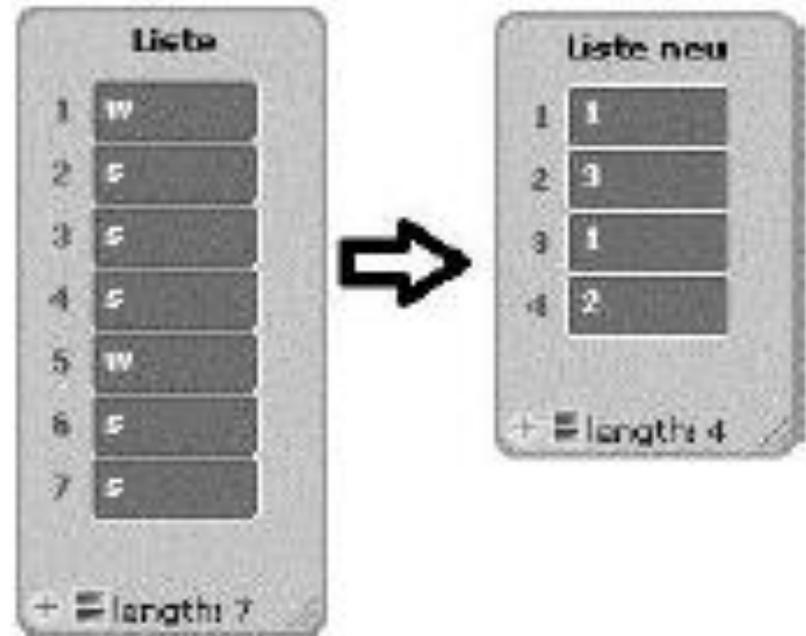
Aufgabenstellung:

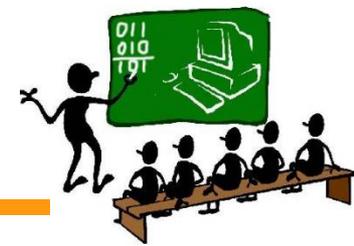
Implementiert ein Programm, das eine Liste mit Einträgen *w* und *s* so zusammenfasst, dass in einer neuen Liste nur die Anzahlen stehen.

Beispiel 1:



Beispiel 2:



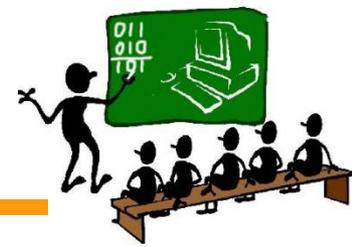


Aufgabe Gruppe 4: „Gescannten Code zusammenfassen – Teil 2“

Ziel: Umwandlung der Ziffernfolge des Strichcodes in die Ziffernfolge der European Article Number (EAN).

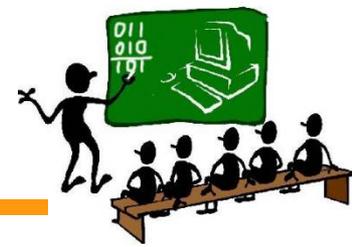
Die vom Strichcode-Scanner aufgenommenen Daten wurden inzwischen so weit verarbeitet, dass sie nun in die zugehörigen Ziffern übersetzt werden können. In der Datei xxx findet ihr eine Liste, in der sich für jede Ziffer jeweils vier Einträge befinden, die den Strichbreiten des zugehörigen EAN-Codes entsprechen.

Ziffer	Listeneinträge	Ziffer	Listeneinträge
0	3211	5	1231
1	2221	6	1114
2	2122	7	1312
3	1411	8	1213
4	1132	9	3112



Zweck steht deutlich im Vordergrund und nicht das innerinformatische Problem

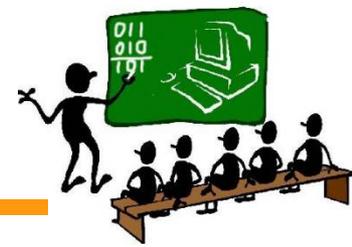
- Motivation
- Dekontextualisierung
- Binnendifferenzierung (Teilgruppen mit unterschiedlichem/-r Schwierigkeitsgrad /Offenheit)



Kontextbezogene Aufgaben – Medizinische Informatik, Strichcode-Scanner, Zeichenprogramm von Kerstin Strecker, LOGIN Heft 176 / 177 (2013), Berlin 2014

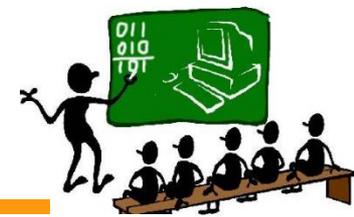
Entwickeln Sie Aufgabenideen ausgehend von einem Kontext. Notieren Sie auf den Karten als Stichworte:

- Kontext
- Informatischer Inhalt
- Jahrgangsstufe



Qualitätskriterien für Aufgaben – Leitfragen zur Aufgabenentwicklung

- Kontextorientierung
- Gestufte Anforderung / Offenheit / Individualisierung und Differenzierung
- Art der Aufgabentypen
- Bildungsstandards – Inhaltsbereiche / Prozessbereiche
- Nachhaltigkeit
- Handlungskompetenz
- Selbstständigkeit
- Formulierung und Strukturierung

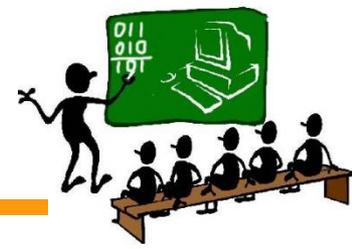


App zur Überwachung von Smartphone-Kosten – Aufgabenidee

Bei monatlichen Abrechnungen von Smartphone-Verträgen haben schon viele Nutzer böse Überraschungen in Form von hohen Gebühren erlebt. Aus diesem Grunde sind Apps zur Überwachung der Kosten eines Smartphones beliebt.

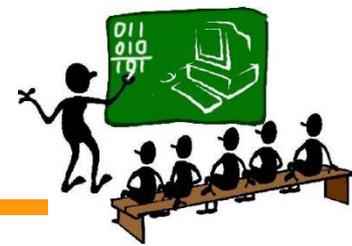
- 1) Implementiere eine Klasse, mit der die Anzahl sämtlicher telefonierter Minuten gespeichert und der aktuelle Wert ausgegeben werden kann. Die Klasse soll eine Methode *gesprachVerbuchen(minuten)* enthalten, die nach einem geführten Telefonat die Summe der telefonierten Minuten aktualisiert und eine Warnmeldung ausgibt, sobald 60 Minuten überschritten werden.
- 2) Erweitere die Klasse aus 1) derart, dass auch die Anzahl abgeschickter Kurznachrichten (SMS) und das übertragene Datenvolumen überwacht werden.

Aufgaben weiterentwickeln



Im Artikel „Aufgabenkultur im Schulfach Informatik“, die Einführung des Themenhefts LOGIN Heft 176 / 177, werden Leitfragen zur Weiterentwicklung von Aufgaben vorgestellt.

Erhöhen Sie die Qualität der vorgestellten Aufgabe, in dem Sie die Teilaufgaben umformulieren und weitere Teilaufgaben hinzufügen. Formulieren Sie die Aufgabenstellung schriftlich. Das Ziel jeder Veränderung müssen Sie mündlich begründen können. (Zeit 15 Minuten)

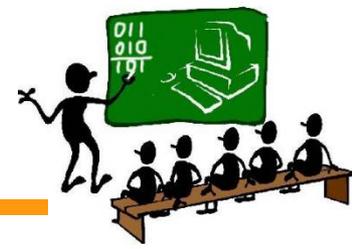


Diagnose – ein wichtiger Baustein im Lehr- und Lernprozess

Kriterien für die Einschätzung der Eignung von Diagnoseaufgaben

- **Validität:** Die Aufgabe konzentriert sich auf die Kompetenzen, zu denen man Informationen gewinnen möchte, und überlagert diese nicht durch andere Aspekte.
- **Offene Antwortformate:** Im Mittelpunkt der Diagnose steht die Eigenproduktion von Schülerinnen und Schülern. Daher müssen diese sich jenseits von vorgegebenen Antwortmöglichkeiten selbst äußern.
- **Individuelle Wege:** Um etwas über Schülervorstellungen erfahren zu können, müssen individuelle Wege und Antworten möglich sein. Dazu sollen Aufgaben offen und differenzierend sein.

aus: BüchtemLeuders, 2005, S. 173

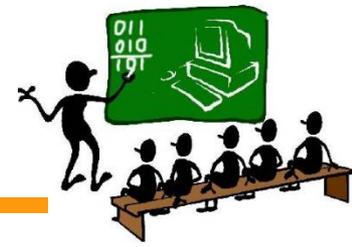


verfahrens- versus **verstehensorientierte** Aufgaben

- Warum funktioniert ein konkretes Verfahren in einer bestimmten Situation?
- Was versteht man unter einem gewissen Begriff?

Typische Aufgabenbeispiele dazu sind

- Erkläre an einem kleinen Beispiel, warum bei Datenbankabfragen über mehrere Tabellen die Daten verschiedener Tabellen durch eine Bedingung verknüpft werden müssen (natürlicher Verbund).
oder
Erkläre die Funktionsweise des gegebenen Quelltextes.
- Erläutern Sie an Hand eines Beispiels den Begriff Parameter knapp aber genau.



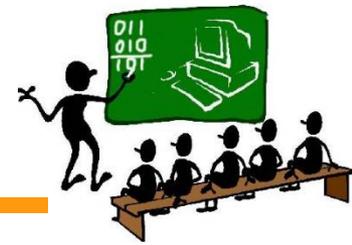
verfahrens- versus verstehensorientierte Aufgaben

- c) Welches Modell (z.B. Klassen- oder Zustandsdiagramm) beschreibt ausgehend von einer Problemanalyse vorgegebene Sachverhalte bzw. Abläufe um eine Realisierung mit einem Informatiksystem zu ermöglichen?

Typische Aufgabenbeispiele dazu sind

- c) Ein Haarföhn hat zwei Schiebeschalter (siehe Bild 4). Der linke regelt die Wärme und hat 2 Positionen (warm, heiß). Der Rechte hat drei Positionen und steuert die Luftmenge (aus, gering, stark - entsprechend der Beschriftung 0, I, II). Erstellen Sie ein v beschriftetes Zustandsdiagramm. Gerne dürfen Sie Abkürzungen verwenden, müssten diese jedoch kurz erklären.



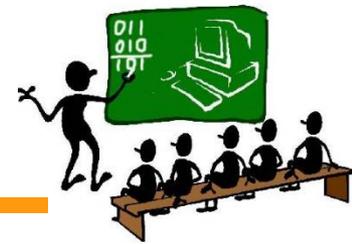


verfahrens- versus verstehensorientierte Aufgaben

- d) Welches Verfahren wähle ich aus, um die gegebene Aufgabenstellung zu lösen?

Typische Aufgabenbeispiele dazu sind

- d) Seit kurzem arbeitet Robot Karol als Bodenleger. Sein Meister hat im Moment viele Aufträge rechteckige Räume mit quadratischen Parkett aus Rotbuchenholz (seine Ziegel) auszulegen. Er empfiehlt Karol dabei immer in der nordwestlichen Ecke des Raumes mit Blickrichtung nach Süden zu beginnen.
- i) Entwirf für Karol eine Anweisung, mit der er die Reihe vor sich bis zur Wand mit Parkett auslegt.
- ii) Entwirf eine Anweisungsfolge, mit der Karol jeden beliebigen leeren Raum vollständig parkettiert.



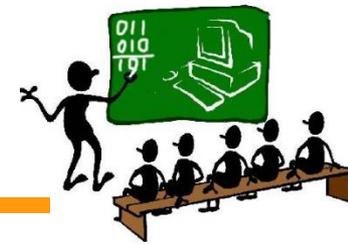
Diagnosepotenzial einer Aufgabe durch offene Antwortformate erhöhen

Aufgabe „Fußgängerampel“ – Variante 1a: Multiple Choice

In der Klasse *FussgaengerAmpel* gibt es die Methode *umschalten()*. Der Aufruf der Methode soll bewirken, dass die Ampel von Rot auf Grün bzw. von Grün auf Rot umschaltet. Kreuzen Sie bei jedem der gegebenen Quelltexte an, ob diese Funktionalität erfüllt wird (richtig) oder nicht (falsch). Falsch gesetzte Kreuze geben Punktabzug.

Quelltext 1: richtig falsch

```
void umschalten(){
    if (ampelphase == "gruen"){
        ampelphase = "rot";
        lampeOben.farbeSetzen("rot");
        lampeUnten.farbeSetzen("schwarz");
    }
    if (ampelphase == "rot"){
        ampelphase = "gruen";
        lampeOben.farbeSetzen("schwarz");
        lampeUnten.farbeSetzen("gruen");
    }
}
```



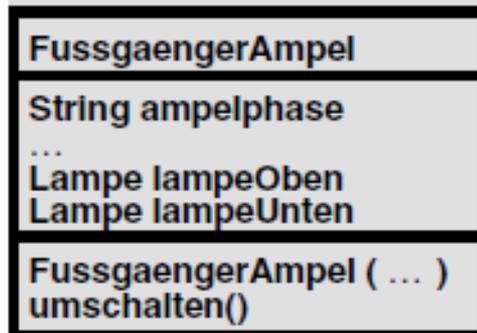
Diagnosepotenzial einer Aufgabe durch offene Antwortformate erhöhen

Aufgabe „Fußgängerampel“ – Variante 2: Von der Funktionalität zum Quelltext

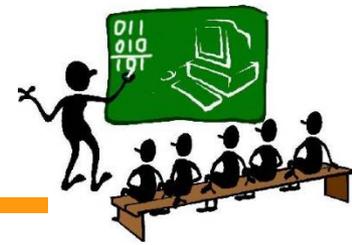
Implementieren Sie in der Klasse *FussgaengerAmpel* (siehe Klassendiagramm im Bild) die Methode *umschalten()*. Der Aufruf dieser Methode soll bewirken, dass die Ampel von Rot auf Grün bzw. von Grün auf Rot umschaltet.

Hinweise:

Der Zustand der Ampel wird im Attribut *ampelphase* mit den Werten „rot“ bzw. „gruen“ gespeichert. Die referenzierten Objekte *LampeOben* bzw. *LampeUnten* sorgen für eine grafische Anzeige. Über die Methode *FarbeSetzen(String neueFarbe)* der Klasse *Lampe* lässt sich die Anzeige ändern.



Klassendiagramm der Klasse *FussgaengerAmpel*.



Diagnosepotenzial

erhöhen

Aufgabe „Fußgängerampel“ – Variante 3:

Screenshot mit syntaktischen und semantischen Fehlern

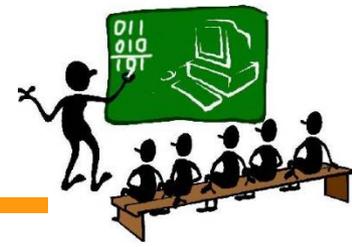
In der Abbildung ist die Methode *umschalten* der Klasse *FussgaengerAmpel* mit einer Fehlermeldung zu sehen.

- i) Erkläre knapp die Ursache des angezeigten Fehlers bei der Übersetzung des Quelltextes.
- ii) Sind Fehler entsprechend i) verbessert und der Quelltext übersetzt, arbeitet die Methode *umschalten()* nicht korrekt. Erkläre knapp die Ursache und verbessere mit Farbe (nicht Rot) direkt im Quelltext.

```
void umschalten() {
    if (ampelphase = "gruen") {
        ampelphase = "rot";
        lampeOben.FarbeSetzen("rot");
        lampeUnten.FarbeSetzen("schwarz");
    }
    if (ampelphase = "rot") {
        ampelphase = "gruen";
        LampeOben.FarbeSetzen("schwarz");
        LampeUnten.FarbeSetzen("gruen");
    }
}
```

incompatible types

Aufgabe selbst entwickeln



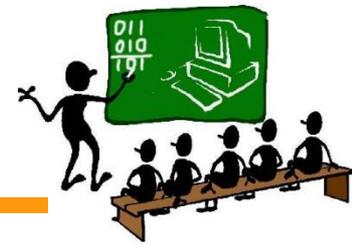
Entwickeln Sie eine "gute" Aufgabe

Möglichkeiten der Vorgehensweise:

Sie entwickeln selbst eine Aufgabe neu oder

Sie nehmen als Ausgangspunkt eine Aufgabe aus einem Schulbuch, dem Internet oder einer anderen Quelle und entwickeln diese weiter.

Aufgabe selbst entwickeln



Möglichkeiten der Vorgehensweise:

Sie entwickeln selbst eine Aufgabe neu oder

Sie nehmen als Ausgangspunkt eine Aufgabe aus einem Schulbuch bzw. dem Internet und entwickeln diese weiter.