



Stufen	5 – 6	leicht	mittel	schwer
Stufen	7 – 8	leicht	mittel	schwer
Stufen	9 – 10	leicht	mittel	schwer
Stufen	11 – 13	leicht	mittel	schwer

Gruppenarbeit

Für eine Gruppenarbeit bildeten die Schüler einer Klasse vier Gruppen. Alle Gruppen teilten ihre Arbeit in einzelne Aufgaben auf. Drei Gruppen konnten alle ihre Aufgaben erledigen, aber eine Gruppe wurde nicht fertig. Was ist passiert?

Die schlauesten Schüler, Ada und Charles, haben die vier Gruppen analysiert. Sie fanden heraus, dass die meisten Gruppenmitglieder auf andere warten mussten, bevor sie mit ihrer eigenen Aufgabe beginnen konnten. Ada und Charles haben für jede Gruppe eine Skizze gezeichnet, die sich auf das Wesentliche konzentriert.

Ein Kreis stellt eine Person dar. Ein Pfeil von Person 1 nach Person 2 bedeutet, dass Person 1 ihre Aufgabe erledigt haben musste, bevor Person 2 mit ihrer Aufgabe beginnen konnte.

Welches Bild entspricht der Gruppe, die nicht fertig wurde?



Antwort D ist richtig:

Die Skizzen stellen Abhängigkeitsgraphen für die Aufgaben der vier Gruppen dar. Die Gruppenmitglieder werden blockiert, wenn ein Zyklus (ein Rundgang) vorhanden ist. Dann kann keiner im Zyklus mit der Aufgaben beginnen, weil jeder auf den Vorgänger wartet. Nur der Graph D enthält einen solchen Zyklus.

Das ist Informatik!

Die meisten Computersysteme erledigen verschiedene Aufgaben scheinbar nebeneinander her. Ein Laptop kann zugleich Musik abspielen, E-Mails abrufen und die Festplatte auf Viren prüfen. Auf einem Smartphone kann man ein Spiel spielen und erhält trotzdem zwischendurch seine SMS und Telefonanrufe. Manche solcher „Prozesse“ können aber voneinander abhängen. Wenn ein Dokument geöffnet werden soll, muss der Editor-Prozess warten, bis der Betriebssystem-Prozess die entsprechenden Daten von der Festplatte kopiert hat. Was passiert aber, wenn zugleich ein anderer Prozess auf die Festplatte zugreifen will? Warten beide Prozesse höflich, bis der andere seine Daten hat?

Die Informatik hat sehr intensiv erforscht, wie solche Deadlocks (Verklemmungen) vermieden werden können. Wer programmiert, muss sicherstellen, dass niemals zwei oder mehr seiner Prozesse aufeinander warten müssen.

<<http://de.wikipedia.org/wiki/Philosophenproblem>>