# Teil I: Hinweise zur Unterrichtskonzeption

## Inhalt


1 Vorwissen der Adressaten ..... 2
2 Unterrichtsmethode ..... 2
3 Kapitelaufbau ..... 2
4 Aufgaben ..... 3
5 Weitgehende Unabhängigkeit von einer Programmiersprache ..... 3
6 Adaptionsmassnahmen ..... 3
7 Evaluationstest Teil 1 (Fachwissen) ..... 4

# 1 Vorwissen der Adressaten

Diese Unterrichtssequenz setzt die Beherrschung elementarer Programmiertechniken mit C# voraus:

- Umgang mit einer Entwicklungsumgebung
- Formularentwurf
- Datentypen und Variablen
- Kontrollstrukturen
- Arrays
- Prozeduren und Funktionen
- Eingabe von / Ausgabe auf Windows-Komponenten
- Rekursion

Ausserdem werden Kenntnisse in einem Tabellenkalkulationsprogramm (z.B. Excel) benötigt:

- Diagrammerstellung
- Logarithmische Skalen

# 2 Unterrichtsmethode

Die vorliegende Unterrichtssequenz eignet sich in dieser Form zum Selbststudium. Für die Bearbeitung der einzelnen Kapitel (inkl. Aufgaben und allenfalls Zusatzaufgaben) ist folgender Zeitplan vorgesehen.

| | |
|---|---|
| Kapitel 1: | 1 Lektion |
| Kapitel 2: | 3 Lektionen |
| Kapitel 3: | 2 Lektionen |
| Kapitel 4: | 2 Lektionen |
| Kapitel 5: | 1 Lektion |
| Evaluationstest Teil 1 (Fachwissen) | 1 Lektion |
| Evaluationstest Teil 2 (Umfrage) | 15 Minuten |

Der Ablauf ist so vorgesehen: Die SchülerInnen bekommen den Stoff kapitelweise, zunächst ohne Lösungen. Nach Ablauf der vorgesehenen Zeit bekommen sie Zugang zu den Lösungen dieses Kapitels. Anschliessend erhalten sie das nächste Kapitel. Die Idee hinter diesem Vorgehen ist es, die SchülerInnen dazu anzuhalten, eine bestimmte Zeit lang selbständig einem Thema zu arbeiten, bevor die Musterlösung beigezogen werden kann.

# 3 Kapitelaufbau

Die meisten Kapitel folgen mehr oder weniger dem gleichen Aufbau: Zunächst wird jedes neue Sortierverfahren mit einer kurzen Geschichte des Mistkäfers Willi eingeführt. In dieser Geschichte entdeckt Willi jeweils die Kernidee des Verfahrens selbständig. Durch diesen Bezug zu einem konkreten Problem im Sinne von Verstehensankern (HG 279ff). sollen die SchülerInnen motiviert und zum Nachdenken angeregt werden. Ausserdem ist das Ziel dieser Geschichten, den SchülerInnen den Respekt vor der Komplexität der einzelnen Verfahren wegzunehmen, insbesondere natürlich bei QuickSort. Es folgen jeweils genauere

Überlegungen zur Funktionsweise des Verfahrens und die Formulierung des Algorithmus. Es wird viel Wert darauf gelegt, jede Idee zuerst gemeinsam und dann als Übungsaufgabe durchzuspielen. Nebst dem Kennenlernen der einzelnen Sortierverfahren soll durch diesen Aufbau auch das allgemeine algorithmische Denken geschult werden.

# 4 Aufgaben

Jedes Kapitel wird durch viele Fragen und Übungsaufgaben aufgelockert, welche der Festigung der vermittelten Inhalte dienen. Auf diese Weise sollten die Schüler in der Lage sein, die drei Verfahren sehr gut zu verstehen und auch weitgehend selbständig implementieren zu können.

Durch die Bearbeitung der Programmieraufgaben entsteht im Laufe der Zeit eine ganze "Programm-Suite", welche sämtliche besprochenen Verfahren enthält. Am Ende der Sequenz haben die SchülerInnen also ein selbst erstelltes Programm in den Händen, welches die wichtigsten Erkenntnisse veranschaulicht (durch das fertige Programm) und zusammenfasst (durch den Quelltext).

# 5 Weitgehende Unabhängigkeit von einer Programmiersprache

Die Unterrichtssequenz und auch die Aufgaben sind grösstenteils unabhängig von einer bestimmten Programmiersprache gestaltet. Die Sequenz kann also sehr einfach auf eine andere Programmiersprache (z.B. Delphi oder Java) angepasst werden.

# 6 Adaptionsmassnahmen

Die Zusatzaufgaben können gute Schüler zu einem tieferen Verständnis des Themas führen. Es handelt sich also hier um eine Adaption des Lernziels, wobei das erweiterte Lernziel absichtlich nicht ausformuliert ist.

Durch die Bearbeitung der normalen Aufgaben können sämtliche Lernziele erreicht werden. Die Zusatzaufgaben gehen zum Teil weit über diese Ziele hinaus. So kommen zum Beispiel bei der letzten Zusatzaufgaben völlig andere Themen wie Grafikprogrammierung ins Spiel.

Von guten SchülerInnen wird nicht nur erwartet, dass sie mehr Aufgaben lösen können als die anderen, sondern auch, dass sie dies selbständiger tun können. Die Zusatzaufgaben sind deshalb bewusst offener formuliert und erlauben unterschiedliche Lösungen.

Bei dem Thema Rekursion gibt es sogar eine Weiche. Guten Schülern wird im Rahmen einer Zusatzaufgabe zugetraut, im Kapitel 3 selbständig eine rekursive Variante des SelectionSort-Algorithmus zu entwickeln. Diese Schüler sollten dann im Kaptitel 4 in der Lage sein, den ganzen, ebenfalls rekursiven QuickSort-Algorithmus selber zu entwickeln, während den anderen Schülern der schwierige Teil mit der Rekursion vorgegeben wird.

# 7 Evaluationstest Teil 1 (Fachwissen)

Der Test enthält zu den Kapiteln (1) bis (4) je eine Aufgabe. Aufgabe (1) hält sich nicht exakt an ein vorgegebenes Lernziel: Es wird nicht die Erklärung eines Begriffs (hier "Algorithmus") verlangt, sondern dessen Anwendung. Aufgaben (2) bis (4) halten sich hingegen exakt an die vorgegebenen Lernziele. Alle Aufgaben besitzen die Taxierungsstufe K3 (Anwendung) (*Frey Allgemeine Didaktik, Kap. 26: Taxieren*).