

Bei einer anschließenden Ausgabe müssen wir ebenfalls wieder jede der 24 Variablen einzeln ansprechen. Wir haben also über 72 Zeilen alleine daraufhin verschwendet, die Texteinträge eines Tages zu verwalten. Wenn wir nun von einer Zahl von 365 Tagen pro Jahr (plus Schaltjahr) ausgehen und unseren Kalender fünf Jahre lang verwenden wollen, erhalten wir eine Programmlänge von weit über 130000 Zeilen! Eine solche Codelänge für ein einfaches Kalenderprogramm ist natürlich völlig untragbar.

Wenn wir das Problem näher analysieren, so stellen wir einen Hauptansatzpunkt für eventuelle Verbesserungen fest: Unsere Variablen `termin_0` bis `termin_23` sind alle vom Typ (`String`) und repräsentieren ähnliche Inhalte (Termine). Auch namentlich unterscheiden sie sich nur durch eine nachstehende Ziffer – wir wollen diese im Folgenden als **Index** bezeichnen. Wenn wir also eine Möglichkeit fänden, die verschiedenen Variablen nur durch diesen Index anzusprechen, könnten wir uns sämtliche Fallunterscheidungen ersparen.

Eine Möglichkeit eröffnen uns die so genannten **Felder**. Mit ihrer Hilfe werden wir Werte wie in einer „Tabelle“ anlegen, sodass wir über eine ganze Zahl (den Index) Zugriff erhalten.

### 5.1.1 Was sind Felder?

Felder (englisch: **arrays**) gestatten es, mehrere Variablen (in unserem Beispiel `String`-Variablen) über einen gemeinsamen Namen anzusprechen und lediglich durch einen **Index** (einen ganzzahligen nichtnegativen Wert) zu unterscheiden. Alle diese indizierten Variablen haben dabei *den gleichen Typ*. Die Variablen selbst werden als Komponenten des Feldes bezeichnet. Der Index, der zum Unterscheiden und zum Ansprechen der Komponenten dient, ist vom Typ **int**, wobei nur Werte größer oder gleich 0 zugelassen werden. Man kann sich vorstellen, dass die Zellen eines Feldes aufeinanderfolgend im Speicher des Rechners abgelegt werden. Der Index einer Variablen ergibt sich dabei aus der Position innerhalb des Feldes, von null aufwärts gezählt. Durch dieses strukturierte Ablegen der Daten innerhalb eines Feldes gelingt die Vereinfachung vieler Anwendungen.

Betrachten wir als Beispiel unseren Terminkalender aus Abbildung 5.4. Wie in Abbildung 5.5 dargestellt, können wir uns die vierundzwanzig Stunden des Tages als **Zeilen** einer Tabelle mit vierundzwanzig Einträgen vorstellen. Wir geben **den Tabellenzeilen** einen einheitlichen Namen – in diesem Fall also `termin` – und betrachten die Stunden somit nicht mehr einzeln, sondern in ihrer Gesamtheit.

Wollen wir einen speziellen Eintrag des Feldes `termin` ansprechen, können wir dies wie gewünscht über seinen Index tun. Wollen wir beispielsweise den in der siebten **Zeile** hinterlegten Text auf dem Bildschirm ausgeben (also zu 6 Uhr morgens), können wir dies in Form der Zeile

```
System.out.println(termin[6]);
```

tun. Folgerichtig erhalten wir die Bildschirmausgabe