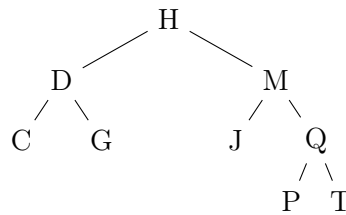


1 Analyse einer Möglichkeit der Traversierung

Betrachten Sie folgenden Baum:



Dieser Baum soll in eine Liste überführt werden, deren Elemente in alphabetischer Reihenfolge vorliegen (C, D, G, H, ...).

Aufgabe 1: Beschreiben Sie das Vorgehen bei dieser Traversierung.

Aufgabe 2: Stellen Sie den Algorithmus in einem Struktogramm dar. Nutzen Sie dabei die Methoden der beiden Datenstrukturen.

Baum:

`gibLinkerTeilbaum(): Baum`

`gibRechterTeilbaum(): Baum`

`gibWurzel(): Inhalt`

`traversiereBaum(baum, liste):`

Liste:

`haengeAn(inhalt)`

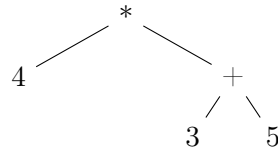
Aufgabe 3: Geben Sie den Namen der Traversierung an und erläutern Sie diese Namensgebung.

5 Minuten
Einzelarbeit!
Fertigstellung bis
10:10 Uhr!

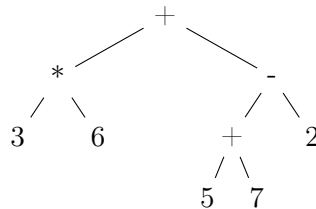


2 Analyse einer Möglichkeit der Traversierung

Mit der Polnischen Notation bezeichnet man die Möglichkeit, eine Rechenaufgabe ohne Klammern eindeutig aufzuschreiben. Dabei wird der Operator immer vor die beiden Operanden geschrieben. Die Aufgabe $4 \cdot (3 + 5)$ wird dann als $\cdot 4 + 3 5$ aufgeschrieben. Als Baum würde es so aussehen:



Der folgende Baum enthält eine Rechenaufgabe, die mit Hilfe einer Traversierung in der polnischen Notation angegeben werden soll.



Aufgabe 1: Beschreiben Sie das Vorgehen bei dieser Traversierung.

Aufgabe 2: Stellen Sie den Algorithmus in einem Struktogramm dar. Nutzen Sie dabei die Methoden der beiden Datenstrukturen.

Baum:

`gibLinkerTeilbaum(): Baum`

`gibRechterTeilbaum(): Baum`

`gibWurzel(): Inhalt`

`traversierePreOrder(baum, liste):`

Liste:

`haengeAn(inhalt)`

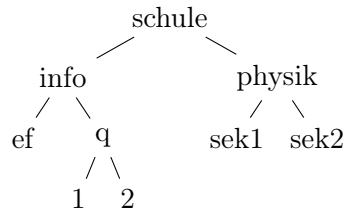
Aufgabe 3: Diese Art der Traversierung nennt man Pre-Order Traversierung. Geben Sie eine Erklärung für diese Namensgebung an.



3 Analyse einer Möglichkeit der Traversierung

Beim Löschen z. B. von Verzeichnissen ist es wichtig, dass jedes Verzeichnis keine Unterverzeichnisse oder Dateien mehr enthält. Diese müssen entsprechend vorher gelöscht werden. Daher muss die Reihenfolge bestimmt werden, in der Verzeichnisse gelöscht werden können. Nehmen Sie dafür folgenden Baum als Beispiel:

5 Minuten
Einzelarbeit!
Fertigstellung bis
10:10 Uhr!



Aufgabe 1: Beschreiben Sie das Vorgehen bei dieser Traversierung.

Aufgabe 2: Stellen Sie den Algorithmus in einem Struktogramm dar. Nutzen Sie dabei die Methoden der beiden Datenstrukturen.

Baum:

`gibLinkerTeilbaum(): Baum`

`gibRechterTeilbaum(): Baum`

`gibWurzel(): Inhalt`

`traversierePostOrder(baum, liste):`

Liste:

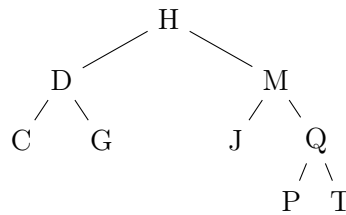
`haengeAn(inhalt)`

Aufgabe 3: Diese Art der Traversierung nennt man Post-Order Traversierung. Geben Sie eine Erklärung für diese Namensgebung an.



4 Analyse einer Möglichkeit der Traversierung

Betrachten Sie folgenden Baum:



Dieser soll in eine Liste überführt werden, bei denen die Elemente in alphabetischer Reihenfolge von hinten nach vorne vorliegen (T, Q, P, M, ...).

Aufgabe 1: Beschreiben Sie das Vorgehen bei dieser Traversierung.

Aufgabe 2: Stellen Sie den Algorithmus in einem Struktogramm dar. Nutzen Sie dabei die Methoden der beiden Datenstrukturen.

Baum:

`gibLinkerTeilbaum(): Baum`

`gibRechterTeilbaum(): Baum`

`gibWurzel(): Inhalt`

Liste:

`haengeAn(inhalt)`

`traversiereReverseInOrder(baum, liste):`

Aufgabe 3: Diese Art der Traversierung nennt man Reverse-In-Order Traversierung. Geben Sie eine Erklärung für diese Namensgebung an.

5 Minuten
Einzelarbeit!
Fertigstellung bis
10:10 Uhr!

