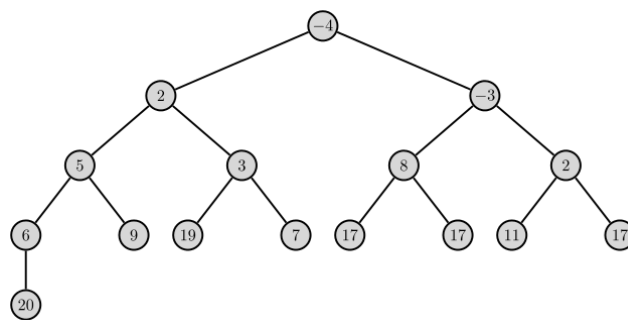
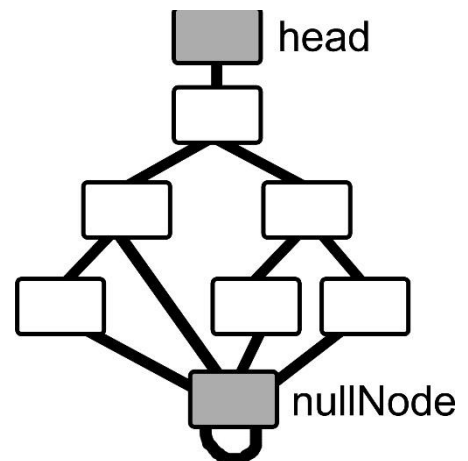


Modul 411

Datenstrukturen und Algorithmen entwerfen und anwenden



```
public static void bubbleSort (ref int[] x)
{
    bool exchanges;
    do
    {
        exchanges = false;
        for (int i = 0; i < x.Length - 1; i++)
        {
            if (x[i] > x[i + 1])
            {
                int temp = x[i];
                x[i] = x[i + 1];
                x[i + 1] = temp;
                exchanges = true;
            }
        }
    }
}
```



Version 4.2
Schuljahr 2020/21

Peter Kaufmann und Roland Bucher

History

Version	Schuljahr	Autor und Beschreibung
1	2015/16	Kap, Peter Kaufmann Skript erstellt
2	2016/17	Kap, Peter Kaufmann Fehlerkorrekturen, leichte Anpassungen, Beispiele an VS 2015 angepasst
3	2017/18	Buo, Roland Bucher Kleine Ergänzungen, weitere Aufgaben und Anpassungen
4	2018/19	Kap Peter Kaufmann Ergänzungen vereinheitlicht, neue Kapitelstruktur, alle Lösungen an VS 2017 angepasst
4.1	2019/20	Kap Peter Kaufmann Fehlerkorrekturen
4.2	2020/21	Kap Peter Kaufmann Fehlerkorrekturen, alle Beispiele und Lösungen an Visual Studio 2019 angepasst

1 Einführung

1.1 Modulidentifikation (Auszug aus dem Modulbaukasten von ict-berufsbildung.ch)

Kompetenz	Problemstellung analysieren und mit geeigneten Datenstrukturen und Algorithmen eine Lösung umsetzen, testen und dokumentieren.
Handlungsziele	<ol style="list-style-type: none"> 1. Für ein gegebenes Problem eine geeignete Datenstruktur definieren und mit den Mitteln einer Programmiersprache, wie Structs, Referenzen / Zeiger und Arrays umsetzen. 2. Ein Problem analysieren und einen geeigneten Algorithmus zur Lösung mit den Grundelementen Zuweisung, Verzweigung und Schleife entwerfen und mit Prozeduren und Funktionen umsetzen. 3. Algorithmen und Datenstrukturen hinsichtlich Speicher- und Zeitkomplexität analysieren und dokumentieren. 4. Ein komplexeres Problem auf kleinere Teilprobleme zurückführen und je nach Problemstellung Iteration oder Rekursion einsetzen. 5. Abstrakte Datentypen, wie Liste, Set, Map etc. und die darauf definierten Operationen kennen und zielgerichtet einsetzen können. 6. Datenstrukturen und Algorithmen mit dem Debugger und weiteren Tools untersuchen und dabei speziell die Situation auf Stack und Heap Analysieren und in geeigneter Form darstellen.

1.2 Handlungsnotwendige Kenntnisse

Handlungsnotwendige Kenntnisse beschreiben Wissens Elemente, die das Erreichen einzelner Handlungsziele eines Moduls unterstützen. Die Beschreibung dient zur Orientierung und hat empfehlenden Charakter. Die Konkretisierung der Lernziele und des Lernwegs für den Kompetenzerwerb sind Sache der Bildungsanbieter.

1.1	Kennt den Unterschied zwischen Wertvariablen und Referenzvariablen / Zeigervariablen.
1.2	Kennt Sprachmittel einer Programmiersprache für das Deklarieren und das Arbeiten mit Werte- und Referenzvariablen.
1.3	Kennt Konzept und Nutzen von verketteten Datenstrukturen.
1.4	Kennt eine Notation für die Darstellung von verketteten Datenstrukturen.
2.1	Kennt eine systematische Vorgehensweise für die Analyse eines Problems.
2.2	Kennt die Grundelemente von Algorithmen und deren Umsetzung in einer Programmiersprache.
2.3	Kennt die Sprachmittel einer Programmiersprache für das Deklarieren und Aufrufen von Funktionen und Prozeduren.
2.4	Kennt Notationen um Algorithmen dazustellen wie z.B. Struktogrammen, PAP usw.
3.1	Kennt Möglichkeiten zur Analyse von Zeit- und Speicherkomplexität von Algorithmen und Datenstrukturen.
3.2	Kennt Komplexitätsklassen von Algorithmen (z.B. lineare, exponentielle Komplexität).
3.3	Kennt Notationen zur Dokumentation der Speicher- und Zeitkomplexität eines Algorithmus.
4.1	Kennt ein Vorgehen bei der Aufteilung eines komplexen Problems in kleinere Teilprobleme.
4.2	Kennt den Aufbau von iterativen Algorithmen.
4.3	Kennt den Aufbau von rekursiven Algorithmen.
4.4	Kennt Vor- und Nachteile sowie geeignete Einsatzgebiete des iterativen und des rekursiven Ansatzes.
5.1	Kennt eine Programmbibliothek, die ADTs wie Liste, Set, Map, etc. zur Verfügung stellt und deren Anwendung.
5.2	Kennt Eigenschaften von einzelnen abstrakten Datentypen.
6.1	Kennt das Speicherlayout für einen Prozess mit den Bereichen Programmcode, globale Variablen, Stack und Heap.
6.2	Kennt die Anordnung von lokalen Variablen und Parametern in einem Stackframe.

1.3 Inhalt des Modulunterrichts

Aus den Handlungszielen bzw. den Handlungsnotwendige Kenntnissen ergibt sich der Inhalt für den Unterricht. Dabei können wir entweder alle Themen kurz ansprechen, da es in den Handlungsnotwendige Kenntnissen immer nur heisst «Kennt...» oder wir vertiefen die wichtigsten Themen und lernen dabei andere eben nur oberflächlich kennen.

Aus diesem Grund haben wir entschieden, die folgenden hauptsächlichen Themen (Blöcke) zu unterrichten:

- Block 01: Einführung
- Block 02: Variablen
- Block 03: Pointer
- Block 04: Strukturen
- Block 05: Verkettete Datenstrukturen (Listen)
- Block 06: Einfache Algorithmen (Sortierung, usw.)

Falls die Zeit reicht, werden wir noch kurz die aktuelle Umsetzung des Gelernten anschauen:

- Block 07: .Net Auflistungsklassen (List, Array usw.)

Das Modul 411 basiert auf den Vorgängermodulen 403 und 404 vom ersten Lehrjahr. Im Modul 403 haben Sie die Programmiersprache C und im Modul 404 C# kennengelernt.

Wir werden in diesem Modul wieder mit C arbeiten. Im Block 07 «.Net Auflistungsklassen» werden wir mit C# die Umsetzung der verketteten Datenlisten in .Net anschauen.

Der ganze Unterricht wird weniger Theorie dafür mehr Praxisanteile (Übungen) umfassen.