
Klausur

aus
Programmierung, Algorithmen und Datenstrukturen
im
Bachelor-Studiengang Wirtschaftsinformatik
12.3.2019

Name	Vorname	UID wi1	PC-Nr
------	---------	------------	-------

- Lesen Sie erst die Angaben aufmerksam, genau und vollständig.
- Die Verwendung von Unterlagen und elektronischen Geräten aller Art, insbesondere Taschenrechner, Mobiltelefonen oder Wechselspeichergeräten, während der Prüfungszeit ist untersagt!
- Sofern nicht anders angegeben ist die für die Beispiele einer Gliederungsebene vorgesehene (Rest-)Punkteanzahl gleichverteilt.
- Die Arbeitszeit beträgt 135 min.

Aufgabe	Punkte
Simple Classes	50
Graphen	15
Bäume und Datenstrukturen	15

Angabe

- Starten Sie **eclipse**
- Erstellen Sie ein Java Projekt (z.B. "Klausur") **im lokalen Documents Ordner**
- Öffnen Sie **Chrome**
- Navigieren Sie zu <https://inf-swe.technikum-wien.at/Assignment>
- Melden Sie sich mit Ihrem LDAP Account an (wi18bxxx)
- Wählen Sie "Upload submission" -> "Upload"
- Gehen Sie zu "BWIVZ-WS18-PAD", "Klausur_190312" und laden Sie die Angaben herunter
- Importieren Sie die heruntergeladene zip Datei in Ihr Java Projekt (File->Import->General->Archive File)
- (Lösen Sie die Beispiele)

Abgabe

- Laden Sie alle Quelldateien (*.java) in der vorgesehenen Abgabe hoch.
- Laden Sie ihre Abgabe erneut herunter und überprüfen Sie die abgegebenen Files.

Achten sie unbedingt darauf Quelldateien (*.java) abzugeben und keine Kompilate!

Viel Erfolg!

1. Simple Classes

Einleitung

In einer Lösung zum Portfoliomanagement werden für **Kursinformationen** folgende Daten erfasst:

- Name (Text, z.B. "Microsoft Corporation")
- Kürzel (4 Zeichen (Großbuchstaben), z.B. "MSFT")
- Datum
- Eröffnungskurs (in \$)
- Schlusskurs (in \$)
- Low (in \$)
- High (in \$)
- Volumen (Anzahl)

Aufgabe

1. [1] Erstellen Sie eine geeignete Klasse `SharePrice` für **die erfassten Größen**.
2. Implementieren Sie folgende Funktionen

- a. [2] `public static void printSharePrice(SharePrice s)`
gibt `s` an der Konsole aus.

Format:

<Name> (<Kürzel>) - <Datum> - <Open>, <High>, <Low>, <Close>. (<Volume>)

(ohne Spitzklammern). Geldbeträge mit 6 Nachkommastellen.

Beispiel:

```
Microsoft Corporation (MSFT) - 2019-03-11 - 110.989998, 112.930000, 110.989998, 112.904999. (12513920)
```

- b. [4] `public static SharePrice scanSharePrice()`
erfasst einen Eintrag an der Konsole.
Validieren Sie die Eingabe des Kürzels mithilfe eines regulären Ausdrucks. Eingabe soll wiederholt werden *bis* gültig. Für alle anderen Werte unterstellen Sie korrekte Eingabe.
- c. [5] `public static SharePriceList add(SharePriceList list, SharePrice s)`
fügt einer Liste einen Eintrag hinzu.
`s` wird in `list` *vorne* angefügt.
- d. [2] `public static void print(SharePriceList list)`
gibt eine Liste an der Konsole aus.
Ein Eintrag pro Zeile.
- e. [7] `public static SharePrice[] toArray(SharePriceList list)`
konvertiert `list` in ein Array.
Erstellt ein neues Array mit allen Einträgen von `list` (und nur jenen, d.h. keine "freien Stellen"). Die Reihenfolge entspricht jener in `list`. `list` wird nicht verändert.
Im Fehlerfall wird `null` zurückgegeben.

- f. [7] `public static SharePrice[] filter(SharePrice[] s, String k)`

filtert Kursdaten nach Name bzw. Kürzel.

Erstellt ein Array mit all jenen (und nur jenen) Einträgen in `s` mit Kürzel oder Name `k`. Die Reihenfolge entspricht jener in `s`. `s` wird nicht verändert.

Im Fehlerfall wird `null` zurückgegeben.

- g. [7] `public static void sort(SharePrice[] shares)`

sortiert `shares` nach Datum.

Adaptieren Sie den gegebenen Code.

```
public static void sort(int numbers[]) {  
    /* i is the first idx of unsorted partition */  
    for (int i = 1; i < numbers.length; i++) {  
        int j = i;  
        /* tmp holds value of next element to insert */  
        int tmp = numbers[i];  
        /* find correct position for tmp */  
        while (j > 0 && tmp < numbers[j - 1]) {  
            /* on the way there shift right */  
            numbers[j] = numbers[j - 1];  
            j--;  
        }  
        numbers[j] = tmp;  
    }  
}
```

- d. [7] `int maxVolume(SharePrice[] s, String k)`

findet den (ersten) Eintrag mit dem größten Volumen von allen Einträgen mit Name oder Kürzel `k`.

Gibt den Index zurück, bzw -1, wenn kein entsprechender Eintrag gefunden wurde oder -2 im Fehlerfall.

3. [7] Erstellen Sie eine kurze Demo-Applikation, welche (unter Verwendung obiger Funktionen)

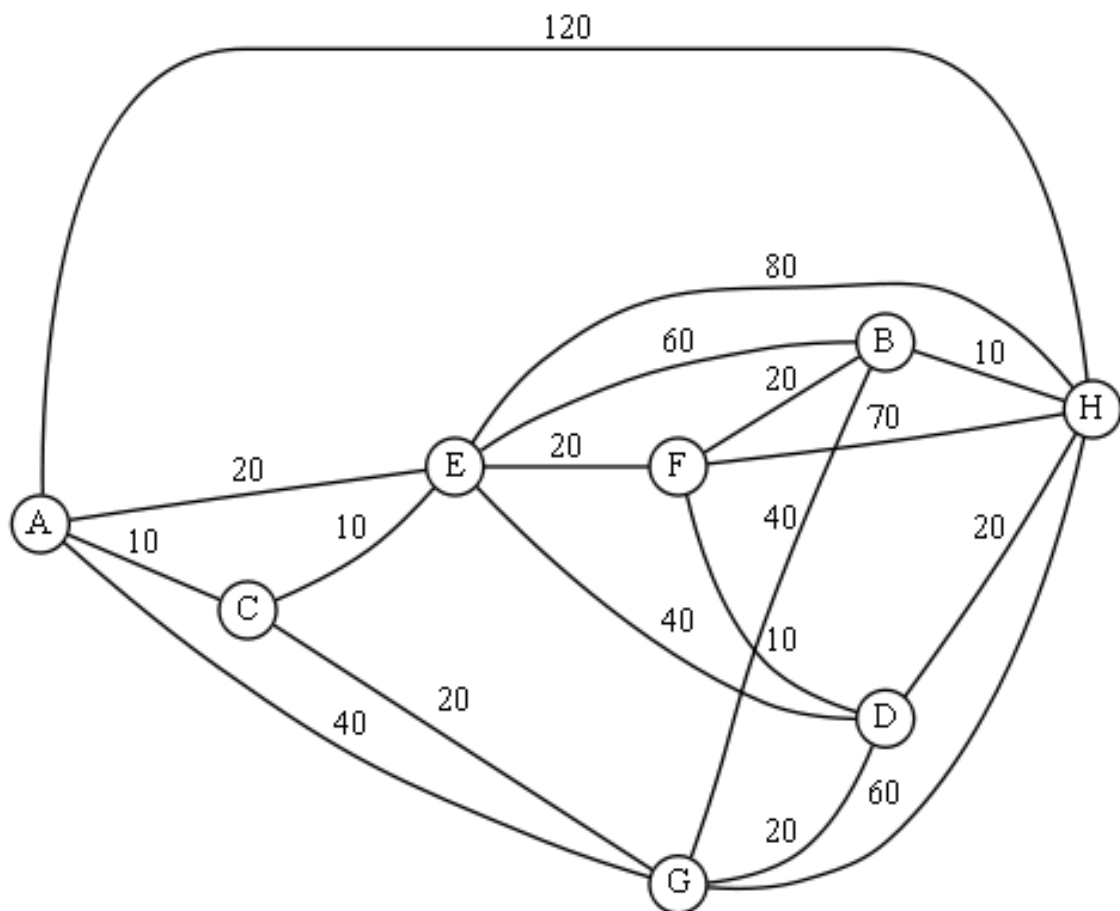
- Eine verkettete Liste mit den Testdaten befüllt
- Einen weiteren `SharePrice` an der Konsole erfasst und der Liste hinzufügt.
- Die Liste ausgibt
- und in ein Array konvertiert
- nach "MSFT" filtert
- und nach Datum sortiert ausgibt (jüngster Eintrag ganz oben).
- Den Tag mit dem größten Volumen für "Apple Inc." ermittelt und ausgibt.

4. [1] Provozieren Sie für eine ihrer Funktionen einen Fehlerfall.

```
//-----
//Testdaten

//(Name, Kürzel, Jahr, Monat, Tag, Open, High, Low, Close, Volume)

("Apple Inc.", "AAPL", 2019, 02, 21, 171.800003, 172.369995, 170.300003, 171.059998, 17249700)
("Microsoft Corporation", "MSFT", 2019, 02, 11, 106.199997, 106.580002, 104.970001, 105.250000, 18914100)
("Microsoft Corporation", "MSFT", 2019, 02, 14, 106.309998, 107.290001, 105.660004, 106.900002, 21784700)
("Microsoft Corporation", "MSFT", 2019, 02, 15, 107.910004, 108.300003, 107.360001, 108.220001, 26606900)
("Microsoft Corporation", "MSFT", 2019, 02, 20, 107.860001, 107.940002, 106.290001, 107.150002, 21607700)
("Microsoft Corporation", "MSFT", 2019, 02, 21, 106.900002, 109.480003, 106.870003, 109.410004, 29063200)
("Apple Inc.", "AAPL", 2019, 02, 15, 171.250000, 171.699997, 169.750000, 170.419998, 24626800)
("Microsoft Corporation", "MSFT", 2019, 02, 27, 111.690002, 112.360001, 110.879997, 112.169998, 21487100)
("Microsoft Corporation", "MSFT", 2019, 02, 12, 106.139999, 107.139999, 105.480003, 106.889999, 25056600)
("Microsoft Corporation", "MSFT", 2019, 03, 07, 111.400002, 111.550003, 109.870003, 110.389999, 25339000)
("Apple Inc.", "AAPL", 2019, 03, 05, 175.940002, 176.000000, 174.539993, 175.529999, 19737400)
("Microsoft Corporation", "MSFT", 2019, 03, 11, 110.989998, 112.930000, 110.989998, 112.904999, 12513920)
("Apple Inc.", "AAPL", 2019, 02, 11, 171.050003, 171.210007, 169.250000, 169.429993, 20993400)
("Microsoft Corporation", "MSFT", 2019, 02, 25, 111.760002, 112.180000, 111.260002, 111.589996, 23750600)
("Apple Inc.", "AAPL", 2019, 02, 12, 170.100006, 171.000000, 169.699997, 170.889999, 22283500)
("Microsoft Corporation", "MSFT", 2019, 02, 13, 107.500000, 107.779999, 106.709999, 106.809998, 18394900)
("Apple Inc.", "AAPL", 2019, 02, 13, 171.389999, 172.479996, 169.919998, 170.179993, 22490200)
("Apple Inc.", "AAPL", 2019, 02, 14, 169.710007, 171.259995, 169.380005, 170.800003, 21835700)
("Apple Inc.", "AAPL", 2019, 02, 25, 174.160004, 175.869995, 173.949997, 174.229996, 21873400)
("Apple Inc.", "AAPL", 2019, 02, 19, 169.710007, 171.440002, 169.490005, 170.929993, 18972800)
("Microsoft Corporation", "MSFT", 2019, 02, 28, 112.040001, 112.879997, 111.730003, 112.029999, 29083900)
("Apple Inc.", "AAPL", 2019, 02, 20, 171.190002, 173.320007, 170.990005, 172.029999, 26114400)
("Microsoft Corporation", "MSFT", 2019, 03, 04, 113.019997, 113.250000, 110.800003, 112.260002, 26608000)
("Microsoft Corporation", "MSFT", 2019, 03, 06, 111.870003, 112.660004, 111.430000, 111.750000, 17687000)
("Microsoft Corporation", "MSFT", 2019, 02, 19, 107.790001, 108.660004, 107.779999, 108.169998, 18038500)
("Apple Inc.", "AAPL", 2019, 02, 22, 171.580002, 173.000000, 171.380005, 172.970001, 18913200)
("Apple Inc.", "AAPL", 2019, 03, 04, 175.690002, 177.750000, 173.970001, 175.850006, 27436200)
("Microsoft Corporation", "MSFT", 2019, 02, 26, 111.260002, 113.239998, 111.169998, 112.360001, 21536700)
("Microsoft Corporation", "MSFT", 2019, 03, 05, 112.250000, 112.389999, 111.230003, 111.699997, 19538300)
("Apple Inc.", "AAPL", 2019, 02, 27, 173.210007, 175.000000, 172.729996, 174.869995, 27835400)
("Apple Inc.", "AAPL", 2019, 02, 28, 174.320007, 174.910004, 172.919998, 173.149994, 28215400)
("Apple Inc.", "AAPL", 2019, 03, 01, 174.279999, 175.149994, 172.889999, 174.970001, 25886200)
("Microsoft Corporation", "MSFT", 2019, 03, 01, 112.889999, 113.019997, 111.669998, 112.529999, 23501200)
("Apple Inc.", "AAPL", 2019, 03, 07, 173.869995, 174.440002, 172.020004, 172.500000, 24796400)
("Apple Inc.", "AAPL", 2019, 03, 11, 175.490005, 179.000000, 175.350006, 178.985001, 23943163)
("Microsoft Corporation", "MSFT", 2019, 02, 22, 110.050003, 111.199997, 109.820000, 110.970001, 27763200)
("Apple Inc.", "AAPL", 2019, 02, 26, 173.710007, 175.300003, 173.169998, 174.330002, 17070200)
("Apple Inc.", "AAPL", 2019, 03, 06, 174.669998, 175.490005, 173.940002, 174.520004, 20810400)
("Microsoft Corporation", "MSFT", 2019, 03, 08, 109.160004, 110.709999, 108.800003, 110.510002, 22810900)
("Apple Inc.", "AAPL", 2019, 03, 08, 170.320007, 173.070007, 169.500000, 172.910004, 23989900)
//-----
```

Figure 1: Graph G_1

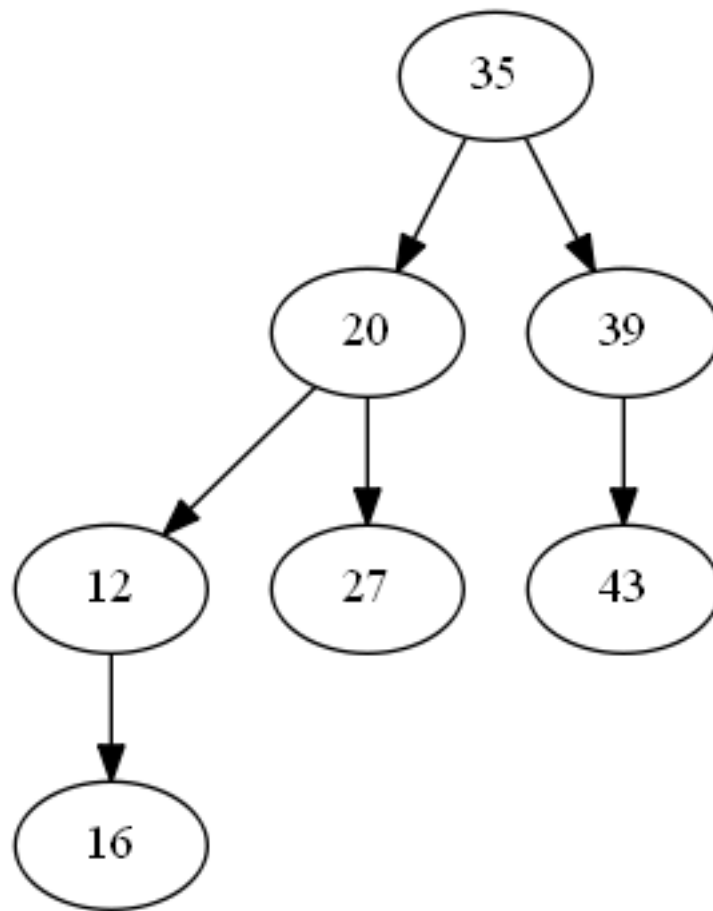
3. Graphen

Gegeben sei G_1 aus Abb. 1.

- [4] Geben Sie für den Graphen aus Abbildung 1 eine Adjazenzmatrix an.
- [8] Finden Sie mittels Dijkstra-Algorithmus die Länge der kürzesten Pfade von Knoten A zu *allen* anderen Knoten. Geben Sie dabei alle Zwischenschritte an!
- [3] Rekonstruieren Sie *alle* kürzesten Pfade von Knoten A zu Knoten H.

ZusatzBlatt2

Geben Sie bei ihren Antworten bitte *immer* die kompletten Beispielnummern an.

Figure 2: BST_1

4. Bäume und Datenstrukturen

Gegeben ist der binäre Suchbaum BST_1 aus Abbildung 2.

- [3] Traversieren Sie den Baum in Postorder Reihenfolge und geben Sie die besuchten Knoten in der entsprechenden Reihenfolge an.
- [3] Fügen Sie die Datensätze mit den Schlüsselwerten 10,36,40 in den Baum ein.
- [4] Löschen Sie aus dem entstehenden Baum nach Aufgabe b die Datensätze mit den Schlüsselwerten 16,43,35. Geben Sie die notwendigen Änderungen nach *jedem* Datensatz an.
- [5] Implementieren Sie in Java eine rekursive Funktion, die die Höhe des binären Suchbaums berechnet. Die Signatur der Funktion könnte wie folgt aussehen:

```
int TreeHeight(TreeNode t)
```

Die Funktion bekommt die Wurzel des Baumes als Parameter übergeben und liefert als Rückgabewert die Höhe des Baumes. Der gegebene Suchbaum hat beispielsweise die Höhe 4 (der linke Teilbaum der Wurzel hat Höhe 3, der rechte Teilbaum Höhe 2)

ZusatzBlatt3

Geben Sie bei ihren Antworten bitte *immer* die kompletten Beispielnummern an.

ZusatzBlatt4

Geben Sie bei ihren Antworten bitte *immer* die kompletten Beispielnummern an.

ZusatzBlatt5

Geben Sie bei ihren Antworten bitte *immer* die kompletten Beispielnummern an.

ZusatzBlatt6

Geben Sie bei ihren Antworten bitte *immer* die kompletten Beispielnummern an.