

Kennzahl: _____

Herbst

Kennwort: _____

2005**46114**Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen**- Prüfungsaufgaben -**Fach: **Informatik (Unterrichtsfach)**Einzelprüfung: **Algorithmen/Datenstrukt./Progr.-meth.**Anzahl der gestellten Themen (Aufgaben): **2**Anzahl der Druckseiten dieser Vorlage: **5**

Bitte wenden!

Thema Nr. 1**Aufgabe 1: Algorithmen und Datenstrukturen**

Sei G ein gerichteter Graph. Die *Transposition* von G hat dieselben Knoten wie G , aber alle Kanten sind entgegengesetzt ausgerichtet.

Bei der Darstellung von Graphen durch *Adjazenzlisten* wird zu jedem Knoten die Liste der benachbarten Knoten angegeben.

- Zeichnen Sie einen gerichteten Graphen mit mindestens 5 Knoten und 10 Kanten und geben Sie dessen Repräsentation durch Adjazenzlisten an!
- Entwerfen Sie ein Verfahren, welches die Transposition eines durch Adjazenzlisten gegebenen Graphen in ebendieser Form bestimmt! Beschreiben Sie das Verfahren detailliert, z.B. durch Angabe von Pseudocode!
- Geben Sie die Komplexität Ihres Verfahrens unter Verwendung der O -Notation als Funktion der Knotenzahl m und der Kantenanzahl n an!
- Beschreiben Sie nun ein Verfahren, welches zu zwei Knoten s, t in einem durch Adjazenzlisten gegebenen Graphen feststellt, ob t von s aus erreichbar ist, und, falls ja, die Länge des kürzesten verbindenden Pfades angibt! Geben Sie auch hier die Zeitkomplexität in der unter c) beschriebenen Form an!

Aufgabe 2: Objektorientierter Entwurf

Im Betriebssystem *Phindows* gibt es vier Arten von Dateien, die jeweils die angegebenen Attribute haben:

- Text: Name, Größe, Anzahl der Zeilen, Verzeichnis. Das Suffix einer Text-Datei ist `.txt`
- HTML: Name, Größe, Anzahl der Zeilen, Titel, Verzeichnis. Das Suffix einer HTML-Datei ist `.htm` oder `.html`
- GIF: Name, Größe, Breite, Höhe, Verzeichnis. GIF-Dateien haben das Suffix `.gif`
- Verzeichnis: Name, Verzeichnis, enthaltene Dateien. Verzeichnisse haben entweder kein Suffix oder eines, welches von den bisher genannten verschieden ist.

Das Attribut "Verzeichnis" gibt jeweils das Verzeichnis an, in dem sich eine Datei befindet.

Bei den folgenden Teilaufgaben können Sie statt des objektorientierten Pseudocodes auch Java verwenden.

Aus dem Pseudocode muss insbesondere die Schnittstelle Ihrer Methoden (also Klassenzugehörigkeit, Anzahl und Typ der Parameter, Typ des Rückgabewertes) eindeutig hervorgehen. Außerdem muss das intendierte Verhalten aller von Ihnen geschriebenen Methoden dokumentiert sein!

- a) Modellieren Sie diese Situation in UML.
- b) Schreiben Sie objektorientierten Pseudocode für eine Methode, die zu gegebenem Namen eine Datei der entsprechenden Art anhand des Suffixes bildet.
- c) Zugriffspfade sind in Phindows durch „|“ getrennte Listen von Dateinamen, etwa „|home|user|test.txt“
Schreiben Sie objektorientierten Pseudocode für eine Methode, die zu einer gegebenen Datei den vollständigen Zugriffspfad als Zeichenkette ausgibt. In Phindows beginnen alle Pfade bei einem Wurzelverzeichnis (im Beispiel home), das sich selbst als Verzeichnis hat.
- d) Schreiben Sie objektorientierten Pseudocode für eine Methode, die in einem gegebenen Verzeichnis und all seinen Unterverzeichnissen nach einer Datei eines gegebenen Namens sucht!

Thema Nr. 2**Aufgabe 1**

Gegeben seien die Schlüsselwerte (3, 5, 11, 8, 4, 1, 12, 7, 2, 6, 10, 9)

- Generieren Sie den binären Suchbaum, der durch sukzessives Einsetzen dieser Werte entsteht. Gehen Sie dabei von einem leeren Baum aus!
- Geben Sie zu den jeweiligen Knoten die zugehörige Balance ein! Ist der Baum ein AVL-Baum?
- Generieren Sie wie in a) erneut einen Suchbaum durch sukzessives Einsetzen der gegebenen Werte und stellen Sie nach jedem Schritt die AVL-Eigenschaft wieder her, so dass beim nächsten Einsetzen ein AVL-Baum zugrunde liegt und auch der Baum am Ende ein AVL-Baum ist.
- Stellen Sie jetzt für obige Schlüsselwertmenge einen B-Baum der Ordnung 3 auf! Gehen Sie schrittweise vor, so dass die Vorgehensweise nachvollziehbar ist!

Aufgabe 2

Die Backus-Naur-Form kann dazu verwendet werden, die Syntax der Kommandozeilenparameter eines Programms zu beschreiben. Als Beispiel wird das Bildkonvertierungstool `convert` herangezogen. Mit diesem können die verschiedensten Operationen auf Bilder, welche durch ihre Dateinamen `file` gegeben sind, ausgeführt werden.

Beschreibung (Auszug):

Version: ImageMagick 6.0.7 11/03/04 Y16 <http://www.imagemagick.org>
Copyright: Copyright (C) 1999-2004 ImageMagick Studio LLC

Usage: `convert {<options>} file [{<options>} file] {<options>} file`
`<options> ::= -antialias | -contrast | -dither | -monochrome |`
`-normalize`

Where options include:

<code>-antialias</code>	remove pixel aliasing
<code>-contrast</code>	enhance or reduce the image contrast
<code>-dither</code>	apply Floyd/Steinberg error diffusion to image
<code>-monochrome</code>	transform image to black and white
<code>-normalize</code>	transform image to span the full range of colors

- Sind folgende Programmaufrufe für die Bilder `bild1.jpg`, `bild2.jpg` und `bild3.jpg` gültig?
Begründen Sie die Antwort!
 - `convert bild1.jpg -antialias bild2.jpg`
 - `convert bild1.jpg`
 - `convert bild1.jpg bild2.jpg -normalize -monochrome bild3.jpg -dither`
 - `convert bild1.jpg bild2.jpg -normalize -monochrome bild3.jpg`
- Wandeln Sie die Backus-Naur-Form in ein Syntax-Diagramm um!

Fortsetzung nächste Seite!

Aufgabe 3:

Gegeben sind folgende Ersetzungsregeln:

$\langle \text{Summe} \rangle ::= \langle \text{Produkt} \rangle \mid \langle \text{Produkt} \rangle + \langle \text{Produkt} \rangle \mid \langle \text{Produkt} \rangle - \langle \text{Produkt} \rangle$
 $\langle \text{Produkt} \rangle ::= \langle \text{Faktor} \rangle \mid \langle \text{Faktor} \rangle * \langle \text{Faktor} \rangle \mid \langle \text{Faktor} \rangle / \langle \text{Faktor} \rangle$
 $\langle \text{Faktor} \rangle ::= \langle \text{Zahl} \rangle \mid (\langle \text{Summe} \rangle) \mid \langle \text{Summe} \rangle$
 $\langle \text{Zahl} \rangle ::= 0 \mid (1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9) \{ 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \}$

- Erstellen Sie ein Syntaxdiagramm zu obigen Ersetzungsregeln!
- Erweitern Sie die Regeln, um das arithmetische Potenzieren darstellen zu können! Verwenden Sie dabei das $^$ Zeichen für die Potenz $a^b \triangleq a^b$. Geben Sie auch die Änderungen am Syntaxdiagramm an!

Aufgabe 4

Eine Rechnung besteht aus mehreren Posten, von denen jeder eine Artikelnummer, Anzahl, Einzelpreis und Gesamtpreis enthält. Eine Rechnung ist an eine Person gerichtet und kann mit unterschiedlichen Zahlungsmethoden bezahlt werden. Zahlungsmethoden sind:

- Barzahlung, ohne weitere Attribute
 - ec-Karte mit BLZ und PIN-Nummer
 - Kreditkarte mit Nummer und Verfallsdatum
- Modellieren Sie den Sachverhalt mit einem ersten UML Klassendiagramm. Geben Sie alle relevanten Beziehungen, Attribute und Methoden an! Begründen Sie Ihre Entwurfsentscheidung! Verwenden Sie hier noch keine Vererbung!
 - Geben Sie ein Objektdiagramm, welches eine Situation beschreibt, in der alle Zahlungsmethoden verwendet werden!
 - Wenden Sie nun Generalisierung (Vererbung) an, um den Entwurf zu verbessern!