

Probe-Klausur im Modul: Grundlagen der Informatik (GDI)

Prüfer: Prof. Dipl.-Ing. Klaus Knopper

Datum: 23.06.2015
Bearbeitungszeit: 180 Minuten

Name, Vorname: _____ Matrikelnr.: _____

Bewertungstabelle:

Aufgabe:	1	2	3	4	5	6	7	8
max.:	10	5	10	25	15	15	10	10
erreicht:								
Σ erreichbar:			Σ erreicht:			Note:		

Erlaubte Hilfsmittel: Bücher, Skripte, Mitschriften, Übungen, Altklausuren, nicht-programmierbarer Taschenrechner, Schreibzeug.

Hinweise zur Bearbeitung dieser Klausur: Kontrollieren Sie bitte, ob Sie alle 10 Aufgabenblätter vollständig erhalten haben. Die Aufgaben lassen sich alle direkt auf dem jeweiligen Aufgabenblatt lösen.
Viel Erfolg!

1 Begriffe der Informatik

1.1 Was versteht man unter einem *Compiler*? (2 Punkte)

Worin unterscheidet er sich von einem *Interpreter*? (2 Punkte)

Antwort: _____

1.2 Was ist ein *Algorithmus*? (4 Punkte)

Antwort: _____

1.3 Was ist ein *Code*? (2 Punkte)

(Zutreffendes bitte ankreuzen. Nur eine Antwort ist richtig.)

- | | |
|--|---|
| ☐ Ein mathematisches Rätsel | ☐ Eine Erfindung von Leonardo DaVinci |
| ☐ Eine Abbildung zwischen Binärwerten und Symbolen oder Zahlen | ☐ Eine Programmiersprache |

2 Grundlagen

2.1 (5 Punkte)

(Bitte bei jeder Antwort ein Kreuz in das richtige Feld einfügen. Richtig angekreuzte Antworten ergeben einen Punkt. Falsch angekreuzte Antworten führen zu einem Punkt Abzug. Nicht angekreuzte Antworten werden nicht bewertet. In der Gesamtwertung trägt die Aufgabe keine negativen Punkte bei.)

	Trifft zu	Trifft nicht zu
Jede Chomsky-Grammatik benötigt mindestens ein Nichtterminal- und ein Terminalsymbol.	☐	☐
Jede von einer Chomsky-Grammatik erzeugte Sprache besteht nur aus Worten, die aus einem einzigen Startsymbol ableitbar sind.	☐	☐
Es gibt in EBNF beschreibbare Sprachen, für die kein Syntaxdiagramm erstellt werden kann.	☐	☐
Der Mechanismus des Überladens betrifft nur <i>Attribute</i> einer Klasse.	☐	☐
Beim Überladen einer Methode kann die Anzahl der Parameter beliebig geändert werden.	☐	☐

3 Zahlendarstellungen

- 3.1 Gegeben sei die folgende Binärzahl (Dualzahl mit der üblichen Interpretation als vorzeichenlose Zahl) 1010101_2 . Wie sieht die dezimale und die oktale Darstellung der Zahl aus? Geben Sie jeweils auch die Schritte zur Berechnung an. (6 Punkte)**

Antwort: _____

- 3.2 Wie sieht die binäre Darstellung der Zahl 41_{10} als 8 Bit Dualzahl aus? (4 Punkte)**

Antwort: _____

4 Grammatik

- 4.1** Überprüfen Sie, ob die folgenden Worte mit den vorgegebenen EBNFs erzeugt werden können und markieren die entsprechenden Stellen mit Ja oder Nein. (12 Punkte)

	$\{ab a\}$	$\{a\}b[a]$	$\{aaab\}[aba]\{aab\}$
aba			
aab			
aaab			
ε			

- 4.2** Gegeben Sei die folgende Grammatik $G = (N, T, P, S)$ mit $T = \{a, b\}$. Bestimmen Sie N und P möglichst minimal so, dass die folgende Sprache erzeugt wird:
 $L(G) = \{w \mid w = ba^{2n}b^n \text{ mit } n \in N_0\}$. (13 Punkte)

5 Algorithmen: Arrays, Funktionen, Zahlendarstellung

5.1 Gegeben sei das folgende Java-Fragment. In diesem werden Dual- und Oktalzahlen als Arrays gespeichert, konvertiert und ausgegeben.

```
public static void main(String[] args) {  
    int[] dual = {1, 0, 1, 1}; //  $8 + 2 + 1 = 11$   
    int[] oktal = {1, 0, 3, 4}; //  $1 \cdot 8^3 + 3 \cdot 8 + 4 = 512 + 24 + 4 = 540$   
    System.out.println("Dualzahl: " + toString(dual));  
    System.out.println("Oktalzahl: " + toString(oktal));  
    System.out.println("Dualzahl Wert: " + toInt(dual, 2));  
    System.out.println("Oktalzahl Wert: " + toInt(oktal, 8));  
    int[] dual2 = octToDual(oktal);  
    System.out.println("Oktalzahl Dualwert: " + toString(dual2));  
}
```

5.1.1 Geben Sie die Implementierung der Methode `toString(...)` an, die ein Array in einen entsprechenden String für die Ausgaben aufbereitet. Im Beispiel oben sollte folgendes ausgegeben werden:

```
Dualzahl: 1011  
Oktalzahl: 1034  
(5 Punkte)
```

- 5.1.2** Geben Sie die Implementierung der Methode `toInt(...)` an, die als Argument neben dem Array die zugrundeliegende Basis erhält und den Wert im Dezimalsystem zurückliefert. Im Beispiel oben sollte folgendes ausgegeben werden:

Dualzahl Wert: 11

Oktalzahl Wert: 540

(5 Punkte)

- 5.1.3** Geben Sie die Implementierung der Methode `octToDual(...)` an, die zu einer gegebenen Array-Repräsentation einer Oktalzahl eine entsprechende Dualzahl erzeugt. Im Beispiel oben sollte folgendes ausgegeben werden:

Oktalzahl Dualwert: 001000011100

(5 Punkte)

7 Programmieren und Programmverständnis

7.1 Gegeben seien die folgenden Variablenbelegungen:

```
int x = 1; int y = 4; int z = 3;
```

Welchen Wert haben die Variablen nach der Ausführung der einzelnen alternativen Zeilen? Der Ausgangswert bei *jeder* Zeile sei die oben angegebene Belegung. (5 Punkte)

	x	y	z
<code>x = x++;</code>			
<code>x *= ++x;</code>			
<code>x = y++ + --z;</code>			
<code>y *= y += z++;</code>			
<code>x -= (z-- - --y);</code>			

7.2 Lassen sich die folgenden Codebeispiele ausführen? Falls nein, erläutern Sie kurz den Fehler. Falls ja - was wird ausgegeben? (5 Punkte)

Code	Ausgabe oder Erklärung
<code>double t = 8.8; int i = t; System.out.println(i);</code>	
<code>int i = 4; float f = i; System.out.println(f);</code>	
<code>int zahl; int zahl2 = zahl; System.out.println(zahl2);</code>	
<code>int[] a1 = {1,2,3,4,5,6}; System.out.println(a1[0]);</code>	
<code>int[] a2 = {1,2,3,4,5,6}; System.out.println(a2[6]);</code>	

8 Objektorientierung

8.1 Ziel sei die Modellierung eines einfachen Systems zur Verwaltung von Prüfungen. Jede Prüfung gehört zu genau einem Modul. Zu jedem Modul kann es mehrere Prüfungen geben. Es muss aktuell zwischen mündlichen und schriftlichen Prüfungen unterschieden werden. Weitere Prüfungsarten können später folgen.

- Jedes Modul soll über einen Namen und eindeutige **Modulnr** verfügen, die von 1000 ab aufsteigend vergeben werden soll.
- Mündliche Prüfungen verfügen noch über einen **Beisitzer**. Dieser wird nach Erzeugung der Prüfung - erst kurz vor dem eigentlichen Termin festgelegt.
- Bei schriftlichen Prüfungen ist die Angabe von **Hilfsmitteln** möglich. Die zulässigen Hilfsmittel werden bei der Erzeugung einer Prüfung angegeben und dürfen danach nicht mehr verändert werden können.

8.1.1 Modellieren Sie in einem ERM-Diagramm alle für die beschriebene Anwendung notwendigen Klassen und ihre Beziehungen zueinander. (4 Punkte)

- 8.1.2 Definieren Sie alle vier für die Verwaltung der Module und Prüfungen notwendigen Klassen in Java. Es müssen nur die Signaturen und Implementierungen der Methoden angegeben werden, die zum Aufbau aller Daten des vollständigen Prüfungsplans erforderlich sind - also keine zur nachträglichen Bearbeitung oder Löschung. Implementieren Sie auch einen Konstruktor für die Klasse Prüfung, der beim Aufruf zwischen einer mündlichen und schriftlichen Prüfung unterscheiden kann. (6 Punkte)**

Summe erreichbare Punkte für Teilaufgabe 8: 10 Punkte

Summe erreichbare Punkte insgesamt: 100 Punkte