

Praktische Informatik 1
(Vorlesung von Prof. Bothe)

WS 05/06

Übungsblatt 2

Abgabe: 14. - 16.11.05 in den Übungen

Die Lösung dieses Übungsblatts soll in Gruppen von je 2 Personen erfolgen.
Die Lösungen sind auf Papier abzugeben. Zu den Aufgaben ist ein Lösungsweg mit anzugeben.
Auf jedem Lösungsblatt sind die Matrikelnummern beider Gruppenmitglieder anzugeben!

Aufgabe 1 (7 Punkte)

Es gilt

$$\begin{aligned} \ln 2 &= \lim_{n \rightarrow \infty} \sum_{i=0}^n \frac{(-1)^i}{i+1} \\ &= 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} \dots \end{aligned}$$

- (5 Punkte) Geben Sie einen Algorithmus in Pseudocode für die Berechnung von $\ln 2$ an, der terminiert, wenn der letzte Summand kleiner als ϵ ist. Eingabe ist ϵ , Ausgabe die $\ln 2$ -Approximation gemäß der Formel sowie der Wert von n in dem Schritt, in dem die Genauigkeits-schranke erreicht war.
- (1 Punkte) Bestimmen Sie die Anzahl der Multiplikations- und Divisionsoperationen in Abhängigkeit von n .
- (1 Punkte) Bestimmen Sie die Anzahl der Additions- und Subtraktionsoperationen in Abhängigkeit von n .

Aufgabe 2 (8 Punkte)

Geben Sie kontextfreie Grammatiken $G_1=[A_1, M_1, s_1, R_1]$ und $G_2=[A_2, M_2, s_2, R_2]$ an, aus denen sich folgende Sätze (und nur diese) ableiten lassen:

- $a^2 b^n a^2$, wobei $n > 1$, (also $aabbaa$, $aabbbbaa$, $aabbbbbaa$, ...)
- a, b, aa, ab, ba, bb .
- Erklären Sie die folgenden Begriffe (mit Hilfe der anderen): *Sprache*, *Grammatik*, *Wort* (*Symbolfolge*), *Symbol*.
- Ist die Pseudocode-Grammatik kontextfrei?

Aufgabe 3 (5 Punkte)

Geben Sie einen allgemeingültigen Algorithmus in Pseudocode an, der alle Primzahlen kleiner als 100 ausgibt.

Pseudocode-Notation

<i>Zuweisung</i>	<i>Variable := Ausdruck</i>
<i>Ausdruck</i>	<i>Variable</i> ganze Zahl reelle Zahl <i>Variable ArithOperator Variable</i> <i>Ausdruck ArithOperator Ausdruck</i> <i>Ausdruck LogOperator Ausdruck</i> <i>Funktion (Ausdruck)</i>
<i>Variable</i>	Bezeichner
<i>Funktion</i>	Bezeichner
<i>ArithOperator</i>	+ - * / DIV MOD
<i>LogOperator</i>	AND OR
<i>VergOperator</i>	= <> < > <= >=
<i>Vergleich</i>	<i>Ausdruck VergOperator Ausdruck</i> <i>Vergleich LogOperator Vergleich</i> NOT <i>Vergleich</i>
<i>Block</i>	BEGIN ... END
<i>Test</i>	IF <i>Vergleich</i> THEN ... ELSE ... ENDIF IF <i>Vergleich</i> THEN ... ENDIF
<i>Schleife</i>	WHILE <i>Vergleich</i> DO ... ENDWHILE REPEAT ... UNTIL <i>Vergleich</i> FOR <i>Variable := Ausdruck</i> TO <i>Ausdruck</i> DO ... ENDFOR
<i>Eingabe</i>	READ (<i>Variable</i>)
<i>Ausgabe</i>	WRITE (<i>Ausdruck</i>)

- Ein Bezeichner ist eine Zeichenfolge, die mit einem Buchstaben beginnt, auf den weitere Buchstaben oder Ziffern folgen können. Ganze Zahlen bestehen nur aus Ziffern, reelle Zahlen enthalten zusätzlich einen Dezimalpunkt.
- Jede verwendete Variable wird als implizit deklariert angenommen.
- In Ausdrücken gilt Punktrechnung vor Strichrechnung. Der Operator / berechnet eine reelle Division, DIV die ganzzahlige Division, MOD wird zur Bestimmung des ganzzahligen Rests einer Division benutzt. DIV und MOD gelten als Punktrechnung. In Vergleichen gilt NOT vor AND und AND vor OR. Gleichrangige Operatoren werden von links nach rechts ausgewertet.
- Leerzeichen und Zeilenumbrüche haben keine Bedeutung. Jede Anweisung (Zuweisung, Block, Test, Schleife, Ein- und Ausgabe) muss durch ein Semikolon abgeschlossen werden.
- Anstelle der drei Punkte (...) darf eine beliebige Anweisungsfolge auftreten.