

Student/in:	Unterschrift:	
 DHBW Duale Hochschule Baden-Württemberg Stuttgart ÜBUNGSKLAUSUR	Fakultät	Technik
	Studiengang:	Angewandte Informatik
	Jahrgang / Kurs :	2013 / 13B
	Studienhalbjahr:	4. Semester
Datum:	20. Mai 2015	Bearbeitungszeit: 90 Minuten
Modul:	TINF2002.3	Dozent: Stephan Schulz
Unit:	Compilerbau	
Hilfsmittel:	Vorlesungsskript, eigene Notizen	
Punkte:	Note:	

Aufgabe	erreichbar	erreicht
1	6	
2	6	
3	10	
4	10	
5	8	
6	7	
7	7	
Summe	54	

1. Sind Sie gesund und prüfungsfähig?
2. Sind Ihre Taschen und sämtliche Unterlagen, insbesondere alle nicht erlaubten Hilfsmittel, seitlich an der Wand zum Gang hin abgestellt und nicht in Reichweite des Arbeitsplatzes?
3. Haben Sie auch außerhalb des Klausorraumes im Gebäude keine unerlaubten Hilfsmittel oder ähnliche Unterlagen liegen lassen?
4. Haben Sie Ihr Handy ausgeschaltet und abgegeben?

(Falls Ziff. 2 oder 3 nicht erfüllt sind, liegt ein Täuschungsversuch vor, der die Note „nicht ausreichend“ zur Folge hat.)

Aufgabe 1 (6 Punkte)

Beschreiben Sie kurz 3 wesentliche Punkte, in denen eine Hochsprache sich von einer maschinennahen Sprache unterscheidet.

Aufgabe 2 (6 Punkte)

Nennen Sie 3 Aspekte der Syntax einer Programmiersprache, die üblicherweise nicht über die Grammatik beschrieben werden, weil sie kontext-sensitiv sind. Wie werden diese Aspekte üblicherweise behandelt?

Aufgabe 3 (10 Punkte)

Das folgende *nanoLang*-Programm enthält 5 Fehler. Identifizieren Sie die einzelnen Fehler und beschreiben Sie, welche Phase des Compilers den Fehler jeweils identifiziert.

```
Integer StrMult(Integer x, String str)
{
    String res;

    res = '';

    while(x>"0")
    {
        res = StrCat(res, str);
        x = x--1;
    }
    return res;
}

Integer main(String count, String str)
{
    print StrMult(StrToInt(count), str);
    print "\n";
    return 0;
}
```

Aufgabe 4 (2+4+4 Punkte)

Betrachten Sie die folgende Grammatik: $G = \langle V_N, V_T, P, E \rangle$ mit $V_N = \{E, V, N, R\}$, $V_T = \{x, y, z, 0, 1, *, +, (,)\}$, und folgenden Produktionen in P :

- $E \rightarrow (E)$
- $E \rightarrow E + E$
- $E \rightarrow E * E$
- $E \rightarrow V$
- $E \rightarrow N$
- $V \rightarrow xR|yR|zR|x|y|z$
- $R \rightarrow xR|yR|zR|0R|1R|x|y|z|0|1$
- $N \rightarrow 0N|1N|0|1$

- a) Geben Sie eine Rechtsableitung für das Wort $(x0 + 10) * xy1$ an.
- b) Generieren Sie eine Ableitung und einen Parse-Tree für das Wort $10 * x + y * z$, der "Punktrechnung vor Strichrechnung" respektiert.
- c) Welche Teile der Grammatik sind kontext-frei und welche sind regulär? Wenn Sie einen Parser für G schreiben müssten, welche Token würde Ihr Parser verarbeiten?

Aufgabe 5 (4+4 Punkte)

Betrachten Sie das folgende *nanoLang*-Programm.

- a) Welche Symboltabellen sind an der Stelle **XXX** angelegt? Geben Sie die Hierarchie und die Einträge an.
- b) *nanoLang* hat statisches Scoping. Welche Variablennamen sind sichtbar? Geben Sie jeweils Name und Typ an.

```
String gstr;
Integer str;

String StrMult(Integer x, String str)
{
    String res;

    res = "";
    str = gstr;

    while(x>0)
    {
        # XXXX
        res = StrCat(res, str);
        x = x-1;
    }
    return res;
}

Integer main(String count, String str)
{
    gstr = str;
    print StrMult(StrToInt(count), "");
    print "\n";
    return 0;
}
```

Fortsetzung

Aufgabe 6 (7 Punkte)

Betrachten Sie das folgende Programmsegment. Geben Sie einen abstrakten Syntax-Baum an, der alle wesentlichen Eigenschaften des Segments wiedergibt. Sie können Knoten mit Operatoren oder Befehlsworten mit ebendiesen beschriften.

```
Integer Fibo(Integer n)
{
    if (n=0)
    {
        return 0;
    }
    if (n=1)
    {
        return 1;
    }
    return (Fibo(n-1)+Fibo(n-2));
}
```

Fortsetzung

Aufgabe 7 (7 Punkte)

Betrachten Sie die folgende Funktion. Wie können Sie die Funktion optimieren, ohne ihr Ergebnis zu verändern? Geben Sie die Optimierungen und das endgültige Programm an.

```
Integer optFun(Integer limit)
{
    Integer sum1;
    Integer sum2;
    Integer i;
    Integer j;
    Integer a;
    Integer b;
    sum1 = 0;
    sum2 = 0;
    i=0;

    while(i<=limit)
    {
        j=0;
        while(j<=limit)
        {
            Integer c;
            a = i*i*j;
            sum1 = sum1+a;
            b=i*i;
            sum2 = sum2+b;
            c = a+b;
            j=j+1;
        }
        i=i+1;
    }
    return sum1/sum2;
}
```

Fortsetzung

– Ende der Klausur –