



KLAUSUR ZUM BACHELORMODUL  
„EINFÜHRUNG IN DIE PROGRAMMIERUNG“ WS 2011/12,  
DR. MAXIMILIAN HADERSBECK  
KLAUSUR AM 06.02.2012

---

VORNAME:

NACHNAME:

MATRIKELNUMMER:

STUDIENGANG:

 B.Sc. Computerlinguistik, B.Sc. Informatik, Magister

Bitte unterstreichen Sie den für Sie zutreffenden Studiengang!

Die Klausur besteht aus **14 Aufgaben**. Die Punktzahl ist bei jeder Aufgabe angegeben. Die Bearbeitungsdauer beträgt **90 Minuten**. Bitte überprüfen Sie, ob Sie ein vollständiges Exemplar erhalten haben.

**Tragen Sie die Lösungen in den dafür vorgesehenen Raum im Anschluss an jede Aufgabe ein. Falls der Platz für Ihre Lösung nicht ausreicht, benutzen Sie bitte nur die ausgeteilten Zusatzblätter!**

Aufgabe	mögliche Punkte	erreichte Punkte
1. Ein-/Ausgabe	3	
2. UNICODE I: Properties	9	
3. UNICODE II: Codierung	2	
4. UNICODE III: Zahlenkonvertierung	2	
5. Subsitus Befehl	9	
6. Reguläre Ausdrücke	10	
7. Subroutine I	8	
8. Subroutine II	10	
9. Rekursion	8	
10. Frequenzliste	22	
11. UNIX-Dateihandling	4	
12. UNIX-Dateihandling	4	
13. UNIX-Secure Copy	3	
14. UNIX-Wortliste	6	
Summe	100	
Note		

**Einwilligungserklärung**

Hiermit stimme ich einer Veröffentlichung meines Klausurergebnisses in der Vorlesung „Einführung in die Programmierung“ vom 06.02.2012 unter Verwendung meiner Matrikelnummer im Internet zu.

Datum: \_\_\_\_\_

Unterschrift: \_\_\_\_\_

NAME:

## 1 Fragen zu utf-8

### Aufgabe 1 Ein-/Ausgabe

1. Welche zwei Methoden gibt es, um STDIN und STDOUT auf utf-8 zu setzen.

(1 Punkt)

2. Was bedeutet die Anweisung use utf8; in einem PERL Programm?

(1 Punkt)

(1 Punkt)  
3 PUNKTE

## 2 Codierung/UNICODE

### Aufgabe 2 UNICODE I: Properties

Was geben die folgenden PERL Fragmente 3.1),3.2) und 3.3) aus, wenn die Zeilen der folgenden Liste mit den regulären Ausdrücken getestet werden.

```
my @lines =("Heute ist der 6. Feb. und",
            "wir schreiben",
            "Klausuri Klausur2",
            "und hoffentlich geht alles gut.");
foreach my $line (@lines)  ...
    ##. hier die Fragmente einsetzen##
}
```

3.1 Fragement:

```
if ($line =~ /(^(\p{Lu}.)+)+\s+(\p{Lu}.)+$/)
    print " $2 $1 \n";
}
```

Ausgabe:

---

---

---

(3 Punkte)

3.2 Fragement:

```
if ($line =~ /(^(\p{L})+\s+(\p{L}+)/)
    print " $2 $1 \n";
}
```

Ausgabe:

---

---

---

NAME:

(3 Punkte)

## 3.3 Fragement:

```
if ($line =~ /(\p{L1})+\s+(\p{L1}+)/) {  
    print "$1 $2\n";  
}
```

Ausgabe:

(3 Punkte)  
9 PUNKTE**Aufgabe 3** UNICODE II: Codierung

In wievielen nachfolgenden Bytes wird die Zeichenkette Straße, bestehend aus den 6 Buchstaben S, t, r, a, ß und e in einer Datei gespeichert, wenn die Datei im Format a) ISO\_LATIN und im Format b) UTF-8 gespeichert ist. Geben Sie Anzahl der Bytes bei a) und b) an:

1. Anzahl der Bytes bei ISO\_LATIN :

(1 Punkt)

2. Anzahl der Bytes bei UTF-8 :

(1 Punkt)

2 PUNKTE

**Aufgabe 4** UNICODE III: Zahlenkonvertierung

Geben Sie die fehlenden Zahlendarstellungen in den entsprechenden Codierung an.

1. Dezimalzahl: 4 Binär: \_\_\_\_\_ Oktal: \_\_\_\_\_ Hexadezimal: \_\_\_\_\_

(1 Punkt)

2. Hexadezimal: A1 Binär: \_\_\_\_\_ Oktal: \_\_\_\_\_ Dezimalzahl: \_\_\_\_\_

(1 Punkt)

2 PUNKTE

### 3 Reguläre Ausdrücke

Was wird bei folgenden substitute Befehlen ausgegeben? Begründen Sie kurz warum?

**Aufgabe 5** Substitute Befehl:

```
1. $string="Das ist Der erste und der letzte, Der die Aufgabe richtig löst.";  
$string=~s/Der./Der/XXX/; print $string;
```

(3 Punkte)

NAME:

2. \$string="Das ist Der erste und der letzte, Der die Aufgabe richtig löst.";  
 \$string=~s/Der.+?Der/xxx/; print \$string;

(3 Punkte)

3. \$string="Das ist Der erste und der letzte, Der die Aufgabe richtig löst.";  
 \$string=~s/Der.+?Der/xxx/ig; print \$string;

(3 Punkte)

9 PUNKTE

**Aufgabe 6** Reguläre Ausdrücke

Sie arbeiten mit Wörtern aus einem deutschen Text, die in der Liste @woerter gespeichert sind. Wie lauten die regulären Ausdrücke (verwenden ausschliesslich(!) UNICODE Properties), die folgende Wörter in der Liste @woerter finden (geben Sie nur den regulären Ausdruck an!).

1. findet alle Wörter, die auf das Suffix "lich" enden. (1 Punkt)
  
2. findet alle Wörter, in denen Datumsangaben mit der Formatierung "Tag.Monat.Jahr" stehen. (2 Punkte)
  
3. findet alle Wörter der Form "Be...ung" mit maximal 10 Buchstaben. (1 Punkt)
  
4. findet alle Wörter die einer Postleitzahl: "DE-80803 München" entsprechen. (1 Punkt)
  
5. findet alle Datumsangaben mit der Formatierung "tt/mm/jjjj". (2 Punkte)
  
6. findet alle Wörter die mindestens ein (großes oder klein geschriebenes) "A" enthalten. (1 Punkt)
  
7. findet alle Wörter nur aus Folgen von Nullen und Einsen bestehen. (1 Punkt)
  
8. findet alle Wörter die nur aus Großbuchstaben bestehen (1 Punkt)

10 PUNKTE

**4 Kleine Programme und Subroutinen****Aufgabe 7** Subroutine I

Schreiben Sie eine Subroutine sub minimum(\$\$) die zwei ganzzahlige Werte als Argument bekommt und den Wert des kleineren Elements zurückgibt.

z.B. \$erg=&minimum(3,6) speichert den Wert 3 im Skalar \$erg

NAME: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

8 PUNKTE

**Aufgabe 8** Subroutine II Schreiben Sie eine Subroutine, die mit einer while Schleife alle Zahlen zwischen 0 und 100 ausgibt. Bei Zahlen, die durch 10 teilbar sind, gibt die Subroutine zusätzlich den Text "durch 10 teilbar" auf der Konsole aus.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

10 PUNKTE

**Aufgabe 9** Rekursion

```
#!/usr/bin/perl
use strict;
{
my ($x,$y,$result);
```

NAME: \_\_\_\_\_

```
print "Enter Zahl1>>";
chomp($x=<>);
print "Enter Zahl2>>";
chomp($y=<>);
$result = &werbinich($x,$y);
print "Result= $result \n";
}
sub werbinich {

my ($m, $n) = @_;

if ($n == 0) {
return 0;

} else {
return $m + &werbinich($m, $n-1);
}

}
```

1. Was berechnet folgendes Programm?

(4 Punkte)

---

2. Was ist das Ergebnis bei der Eingabe 1 und 5 ?

(1 Punkt)

---

3. Was ist das Ergebnis bei der Eingabe 3 und -5 ?

(3 Punkte)  
8 PUNKTE

## 5 PERL-Programm: Erstellen einer Frequenzliste

**Aufgabe 10** Von einer utf-8 Datei sz.txt, in der deutscher Text gespeichert ist, sollen die Anzahl der Wörter und die 5 seltensten Wörter ausgegeben werden, die maximal 5 Buchstaben lang sind.

Die Datei ist zu öffnen, soll zeilenweise gelesen, die Anzahl aller Wörter gezählt werden und die gesuchten Wörter in einen HASH eingetragen werden.

Der HASH soll nach der Häufigkeit der Wörter sortiert werden und die sortierten Keys in einer Liste gespeichert werden. Aus dieser Liste sollen die 5 seltensten Wörter auf dem Terminal ausgegeben werden und in der Datei unimportant.txt gespeichert werden.

Es soll die Anzahl aller Wörter ausgegeben werden.

Tragen Sie in den freigelassenen Raum auf der linken Seite, die den Anweisungen auf der rechten Seite entsprechenden PERL-Programmzeilen ein.

NAME:

- # Startzeile eines PERL-Programms 22 PUNKTE

# PERL soll sich streng verhalten

# Codierungseigenschaften einstellen

# Codierungseigenschaften einstellen

# Programmanfang

# Variablendeclaration (*füllen Sie diese Zeilen erst aus, wenn Sie wissen, welche Variablen benötigt werden*)

# Öffnen der utf8 Datei sz.txt zum Lesen

# Öffnen der utf8 Datei unimportant.txt zum Schreiben

# Zeilenweise einlesen nach \$line mit while

# Zerteilen Sie line an Zeichen, die keine Buchstaben sind (verwenden sie UNICODE Properties) und speichern Sie das Ergebnis in der Liste woerter)

#

# Durchlaufen der Liste, zählen Sie die Wörter und zählen sie die gesuchten Wörter mit maximal 5 Buchstaben im HASH freq

#

#

# while-Schleife zu Ende

# Speichern Sie alle Wörter, die in freq vorkommen, nach Häufigkeit sortiert, in sortierte\_woerter

# Geben Sie am Terminal aus, wieviele Wörter im Text vorkommen

# Geben Sie die fünf seltensten Wörter auf dem Terminal aus und speichern Sie sie in der Ausgabedatei unimportant.txt.

# nach dem Muster (z.B.): über = 1  
# könne = 1  
# ...

# Schliessen Sie geöffneten Dateihandles

#

#

# Programmende

NAME:

## 6 UNIX

**Aufgabe 11** UNIX–Dateihandling

Ausgehend von folgenden Informationen auf Ihrer Konsole sollen sie mit UNIX-Befehlen folgende Aufgabe lösen

```
studiosus@computer:~/projekt>ls -la
insgesamt 128
drwxr-xr-x  2 studiosus student  4096 30. Jan 15:20 .
drwx-----x 251 studiosus student 36864 30. Jan 15:19 ..
-rw-r--r--  1 studiosus student 84905 30. Jan 15:19 foto.jpg
studiosus@computer:~/projekt>
```

1. Wechseln Sie bitte in ihr Homeverzeichnis!

(1 Punkt)

- 
2. Legen Sie dort ein Verzeichnis mit dem Namen backup an!

(1 Punkt)

- 
3. Kopieren Sie foto.jpg in das Verzeichnis backup!

(2 Punkte)

---

4 PUNKTE**Aufgabe 12** UNIX–Dateihandling

Sie haben eine Datei aus dem Internet geladen:

```
studiosus@computer:~/text>ls -la
insgesamt 128
drwxr-xr-x  2 studiosus student  4096 30. Jan 15:20 .
drwx-----x 251 studiosus student 36864 30. Jan 15:19 ..
-rw-r--r--  1 studiosus student 1284905 30. Jan 15:19 textarchiv.tar.gz
studiosus@computer:~/text>
```

Wie lauten die einzelnen UNIX Befehle?

1. Packen sie diese Datei vollständig aus.

(3 Punkte)

- 
2. Lassen Sie sich anzeigen wieviel Speicherplatz in Ihrem Arbeitsverzeichnis(.) noch frei ist

(1 Punkt)

---

---

---

4 PUNKTE

NAME:

**Aufgabe 13** Secure Copy

Erklären Sie kurz den grundlegenden Unterschied zwischen den Befehlen scp und ssh!

(1 Punkt)

Erläutern Sie alle Bestandteile des folgenden Befehlsaufrufs:

```
scp urlaubsbilder.zip studiosus@remote.cip.ifi.lmu.de:privat/
```

Erläutern Sie den Unterschied des folgenden Befehlsaufrufs zum vorherigen:

(1 Punkt)

```
scp urlaubsbilder.zip studiosus@remote.cip.ifi.lmu.de:/privat/
```

(1 Punkt)  
3 PUNKTE**Aufgabe 14** UNIX-Wortliste

Erstellen Sie aus einer Datei "wortliste.txt" (beinhaltet ein Wort pro Zeile) eine Datei "frequenzliste.txt" (Anzahl, Wort; sortiert nach Anzahl) Nutzen Sie dazu die Befehle cat, sort, tr und uniq unter Anwendung von Pipes in einer Befehlszeile.

(6 Punkte)  
6 PUNKTE

NAME:

---

**es folgt ein Schmierblatt... Bitte entfernen**