

## Fehler in der Klausuraufgabenstellung sind mit **rot** markiert

### 1. Fragen zu utf-8:

1. Was bedeutet die Anweisung *use utf8*; in einem Perl Programm?

Die Codierung des Programms wird auf utf8 gesetzt. So sind zum Beispiel Umlaute in Variablenamen erlaubt.

2. Was bedeutet die Anweisung *use strict*; in einem Perl Programm?

Es werden einige syntaktische Überprüfungen eingeschaltet.

3. Wie lautet das *open* Statement in Perl zum Öffnen der utf-8 Datei *sz.txt* zum Schreiben?

```
open(SCHREIBEN, ">:utf8", "sz.txt");
```

### 2. Unicode und Codierung:

Was geben die folgenden PERL Fragmente aus, wenn die Zeilen der folgenden Liste mit den regulären Ausdrücken getestet werden.

```
my @lines = ("mir geht es gut?",
"Die 1. Klausur im Februar 2014",
"ist sehr lange.");
foreach my $line (@lines) ...
##. hier die Fragmente einsetzen##
}
```

```
2.1 while ($line =~ /\{P{L}\}(.{3,})\s/g) {
print " $1 $2 \n";
}
```

```
geht es
Die 1. Klausur im Februar
sehr
```

```
2.2. if ($line =~ /\{p{L}.*\}(\{p{P}\})/) {
print " $1 gefolgt von $2 \n";
}
```

je nachdem wo die Klammer wieder zugeht!

in meinem Fall:

mir geht es gut gefolgt von ?

Die 1 gefolgt von .

Ist sehr lange gefolgt von .

```
2.3. if ($line =~ /\s([^\s]+?)\s(.*)$/) {
print "$1 $2\n";
}
```

Achtung! [^.] bedeutet einfach nur "kein Punkt" → Punkt in eckigen Klammern ist einfach nur Punkt und kein Metazeichen

geht es gut?

Klausur im Februar 2014

sehr lange.

3. In wievielen nachfolgenden Bytes wird das Wort *Öl*, bestehend aus den 2 Buchstaben *Ö* und *l* in einer Datei gespeichert, wenn die Datei im Format 1) ISO-Latin und im Format 2) utf-8 gespeichert ist. Geben Sie Anzahl der Bytes bei 1) und 2) an:

3.1.

2 Byte

3.2.

3 Byte

4. Geben Sie die fehlenden Zahlendarstellungen in den entsprechenden Codierungen an.

Dezimal	Binär	Oktal	Hexadezimal
11	1011	13	B
175	10101111	257	AF

### 3. Reguläre Ausdrücke:

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

5.1. `$string="Sie findet in der Siemensküche sieben Siegel im Teesieb";`  
`$string=~s/Sie.+sie/xxx/; print "$string";`

xxx → Es wird vom ersten Sie bis zur letzten Möglichkeit von sie ersetzt (greedy). Also von Sie bis sie in Teesieb.

5.2. `$string="Sie findet in der Siemensküche sieben Siegel im Teesieb";`  
`$string=~s/sie.+?sie/xxx/ig; print $string;`

xxxmensküche xxx gel im Teesieb → Es wird non greedy und global mit ignore case von sie bis sie ersetzt. Also von Sie bis Siemensküche und von sieben bis Siegel. Nongreedy sucht sich die kleinste Möglichkeit etwas zu ersetzen.

5.3. `$string="Tiger haben gestreifte Haut, nicht nur gestreiftes Fell.";`  
`$string=~ s/gestreift(\p{L}+)/schön$1/; print $string;`

Tiger haben schöne Haut, nicht nur gestreiftes Fell. → es wird gestreift mit der dazugehörigen Endung von schön mit der passenden Endung ersetzt. Dies passiert nur einmal, da kein global gesetzt ist.

6. 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!).

6.1. findet alle Wörter, die mit 2 Vokalen anfangen.

\$wort =~ /^[aeiouäöü]{2}/i

6.2. findet alle Wörter, die mit einem Kleinbuchstaben beginnen und mit einem Punctuationszeichen enden.

\$wort =~ /^p{Ll}.\*p{P}\$

6.3. findet alle Wörter, die mit einem Großbuchstaben, mit heit enden und maximal 10 Buchstaben lang sind.

\$wort =~ /^p{Lu}.{0,5}heit\$

6.4. findet alle Zahlen (auch Kommazahlen und Zahlen mit Tausendertrennzeichen, also z.B. 2,99 oder 12.800).

\$wort =~ /^p{N}|(p{N}+[.,]?)p{N}+)\$/ (kann man drüber diskutieren)

6.5. findet alle Wörter die genau eine Zahl enthalten.

\$wort =~ /^P{N}\*p{N}P{N}\*\$/

6.6. findet alle Wörter mit Buchstaben die alphabetisch zwischen "a" "m" liegen.

\$wort =~ /^[a-m]+\$/i

6.7. findet alle Wörter, die mit einem Konsonanten (Tipp: einem „Nichtvokal“ also) anfangen und höchstens 8 Buchstaben lang sind

\$wort =~ /^[^aeiouäöü].{0,7}\$/i

#### 4. Kleine Programme und Subroutinen:

7. Schreiben Sie eine Subroutine sub anagramm(\$) die eine Zeichenkette als Skalar als Argument bekommt und den Wert 1 zurückgibt, wenn die Zeichenkette ein Anagramm ist, ansonsten den Wert 0. (Sie dürfen die Systemroutine rewind nicht verwenden.) z.B. \$ja=&anagramm("AnnA") gibt den Wert 1 zurück.

**hier ist Palindrom gemeint**

```
sub palindrom($){
    $a=$_[0];
    $a=lc($a);
    my @buchst= split(//,$a);
    my $palindrom = 1;

    for (my $i = 0; $i < length($a)/2; $i++) {
        if ($buchst[$i] ne $buchst[length($a)-1-$i]) {
            $palindrom = 0;
        }
    }
}
```

```

    if ($palindrom == 0) {
        return 0;
    }
    else {
        return 1;
    }
}

```

8. Schreiben Sie eine Subroutine `sub mittelwert($@)`, die mit zwei Argumenten aufgerufen wird. Im ersten Argument ist gespeichert, wie viele Elemente in der Liste sind. Die Liste wird als zweites Element übergeben. Die Subroutine soll den Mittelwert aller Werte aus der übergebenen Liste berechnen und zurückliefern. z.B. `$erg=&mittelwert(3,(2,4,6))` berechnet den Wert 4 im Skalar `$erg`

```

sub mittelwert($@) {
    my ($zahl, @liste) = @_;
    my $a = 0;

    for (my $i = 0; $i < $zahl, $i++) {
        $a = $a + $liste[$i];
    }
    return $a/$zahl;
}

```

### 9. Rekursion

```

#!/usr/bin/perl
use strict;
{
    my (@werte) = (1,34,6,4,9);
    my $result = &werbinich(@werte);
    print "Result= $result \n";
}
sub werbinich (@) {
    my @liste = @_;
    if (scalar(@liste) == 0) {
        return 0;
    } else {
        shift(@liste);
        return 2 + &werbinich(@liste);
    }
}

```

9.1. Welchen Wert gibt das Programm aus?

Result = 10

9.2. Was ist das Ergebnis bei der Liste `my (@werte) = (1,34,6);`?

Result = 6

9.3 Was ist das Ergebnis, bzw. was passiert, wenn in der Subroutine anstatt `shift(@liste)` die Anweisung `pop(@liste)` steht?

Die Liste wird statt von vorne von hinten gekürzt. Die Ausgabe bleibt gleich.

9.4. Was ist das Ergebnis, bzw. was passiert, wenn in der Subroutine anstatt `shift(@liste)` die Anweisung `push(@liste, 1)` steht?

Das Programm kann nicht terminieren. Die Liste wird immer länger und kommt nie zum Abbruchkriterium.

## 5. Erstellen einer Frequenzliste:

10. Es werden die vier häufigsten Wörter in einer Datei gesucht, die mindestens 4 Buchstaben lang sind. Von einer utf-8 Datei `sz.txt`, in der deutscher Text gespeichert ist, sollen die 4 häufigsten Wörter ausgegeben werden, die mindestens 4 Buchstaben lang sind. Die Datei ist zu öffnen, soll zeilenweise gelesen, die Anzahl der gesuchten Wörter gezählt werden und 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 4 häufigsten Wörter auf dem Terminal ausgegeben werden und in der Datei `viele.txt` gespeichert werden. Es soll die Anzahl aller gesuchten Wörter ausgegeben werden. Tragen Sie in den freigelassenen Raum auf der linken Seite, die den Anweisungen auf der rechten Seite entsprechenden PERL-Programmzeilen ein.

```
#!/usr/bin/perl
use strict;
use utf8;
use locale;
{
my ($line, @woerter, $elem, %freq, $zaehler, @sortierte_woerter);
open(LESEN, "<:utf8", "sz.txt") or die "File not found";
open(SCHREIBEN, ">:utf8", "viele.txt");

while ($line = <LESEN>) {
    chomp $line;
    @woerter = split(/\p{L}/, $line);

    foreach $elem (@woerter) {
        if ($elem =~ /\p{L}{0,4}$/) {
            $freq{$elem}++;
        }
    }
}
@sortierte_woerter = (sort{$freq{$b} <=> $freq{$a}} keys %freq);

print (scalar(@sortierte_woerter));

my $zaehler = 0;
```

```

foreach $elem(@sortierte_woerter) {
    if ($zaehler < 5) {
        print "$freq{$elem} = $elem.\n";
        print AUS "$freq{$elem} = $elem.\n";
        $zaehler++;
    }
}

close LESEN;
close SCHREIBEN;
}

```

## 6. Unix:

11. Ordnen Sie die Befehle den Aufgabenbereichen zu:

Prozessverwaltung: ps, top, kill, renice

Netzwerk: ip, netstat, ifconfig

Dateien: du, ls, chmod

12. Was ist hier passiert? Statt "abc äöü" wird "abc Ã¤Ã¶Ã¼" angezeigt. Erklären Sie die Ursache und mögliche Abhilfe!

Die Codierung ist nicht auf utf-8 eingestellt. Codierung muss auf utf-8 eingestellt werden.

13. Gegeben ist ein entfernter Server mit folgenden Daten:

-die IP-Adresse: 2001:db8::1

-der Loginname: chef

-das Passwort: dckskpm!3

-der Dateiname und Pfad: /projekt/wichtig.zip

-der Hostname ist nicht bekannt

Kopieren Sie die angegebene Datei auf Ihren Rechner in Ihr aktuelles Verzeichnis mit Hilfe des Befehls "scp"! Hinweis: IPv6-Adressen sind in vielen Fällen, so auch bei der Verwendung mit scp in eckige Klammern [] zu setzen. Wie lautet der Befehl?

Welche Angabe wird interaktiv abgefragt?

scp chef@[2001:db8::1]: ~/projekt/wichtig.zip .

Passwort wird interaktiv abgefragt.

14. Ein- und Ausgabeumleitung

Sie haben eine Datei namens "dump.sql" erhalten. Diese soll via "stdin" dem Programm "mysql" übergeben werden. Welcher der drei folgenden Aufrufe ist falsch?

1. mysql < dump.sql

2. cat dump.sql | mysql

3. cat dump.sql > mysql

Was passiert bei dem falschen Aufruf?

Version 3 ist falsch, weil hier versucht wird dump.sql in eine Datei zu schreiben statt es in das Programm zu übergeben.

*15. Sie haben einfachen Text (Kodierung utf8) als komprimierte Datei erhalten: text.txt.gz Packen Sie diese Datei aus!*

```
tar -zx text.txt.gz
```

*Erstellen Sie eine Datei "frequenzliste.txt" (Eine Liste mit zwei Spalten Anzahl und Wort, nach Anzahl sortiert) mit Hilfe der Linux-Befehle: cat, sort, tr und uniq unter Anwendung von Pipes in einer Befehlszeile.*

```
cat text.txt | tr " " "\n" | sort -f | uniq -ic | sort -n > ./frequenzliste.txt
```