

Musterlösung zu Blatt 8:

1. Spalten Sie mit split alle Zeilen der Datei wittgenstein.txt in Wörter auf und erzeugen Sie eine Frequenzliste der Wörter.

1.1 Sortieren Sie die Frequenzliste nach der Häufigkeit und speichern Sie die nach der Häufigkeit sortierten keys im Array @alle_sortierten_woerter

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: erzeugt Frequenzliste und speichert nach Häufigkeit in
@alle_sortierten_woerter
use utf8;
use locale;
use strict;
{
    my ($zeile, %lexikon, @woerter, $element, @alle_sortierten_woerter);
    open(ARTIKEL, "<utf8", "wittgenstein.txt") or die "File not found!";
    #gibt Fehlermeldung aus wenn Datei nicht gefunden wurde

    while ($zeile = <ARTIKEL>) {
        chomp($zeile);
        @woerter = split(/[\p{Z}\p{P}]+/, $zeile);

        foreach $element (@woerter) {
            $lexikon{$element]++;
        }
    }

    @alle_sortierten_woerter = sort {$lexikon{$a} <=> $lexikon{$b}} keys
%lexikon;

    foreach $element (@alle_sortierten_woerter) {
        print "Wort $element mit der Häufigkeit $lexikon{$element}\n";
    }

    close ARTIKEL;
}
```

2. Schreiben Sie ein PERL Programm, das alle großgeschriebenen Wörter im Array @alle_sortierten_woerter findet und die Wörter zusammen mit Ihrer Häufigkeit in der Datei "grosse_woerter.txt" speichert.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: speichert große Wörter in grosse_woerter.txt
use utf8;
use locale;
use strict;
{
    my ($zeile, %lexikon, @woerter, $element, @alle_sortierten_woerter);
    open(ARTIKEL, "<:utf8", "wittgenstein.txt") or die "File not found!";
    open(AUSGABE, ">:utf8", "grosse_woerter.txt") or die "File not found!";

    while ($zeile = <ARTIKEL>) {
        chomp($zeile);
        @woerter = split(/[\p{Z}\p{P}]+/, $zeile);

        foreach $element (@woerter) {
            $lexikon{$element]++;
        }
    }

    @alle_sortierten_woerter = sort {$lexikon{$a} <=gt $lexikon{$b}} keys %lexikon;

    foreach $element (@alle_sortierten_woerter) {
        if ($element =~ /^[^\p{Lu}]$/) { #nur grosser Anfangsbuchstabe
            #if ($element =~ /^[^\p{Lu}+$/ { #Wort mit nur Großbuchstaben
                print AUSGABE "Das Wort $element mit der Häufigkeit
$lexikon{$element}.\n";
            }
        }
    }

    close ARTIKEL;
    close AUSGABE;
}
```

3. Wie lauten die regulären Ausdrücke, die folgende Wörter im Array @alle_sortierten_woerter finden (geben Sie nur den regulären Ausdruck an!). Testen Sie Ihre Ausdrücke mit Ihrem PERL-Programm aus

3 .1. alle Wörter mit 3 Buchstaben

\$element =~ /^[p{L}{3}]\$/

3 .2. alle Wörter in denen ein Selbstlaut vorkommt.

\$element =~ /[aeiouäöü]/i

oder:

\$element =~ /^[^aeiouäöü]*[aeiouäöü][^aeiouäöü]*\$/i (genau ein Selbstlaut)

3 .3. alle Wörter mit Umlauten

\$element =~ /[\äöü]/i

3 .4. alle Wörter die mit 'net' enden.

\$element =~ /net\$/

3 .5. alle Wörter die drei bis fünf Buchstaben lang sind und mit einem Vokal beginnen.

\$element =~ /^[aeiouäöü]\p{L}{2,4}\$/i

3 .6. alle Wörter die mit einem Großbuchstaben beginnen und mit dem Suffix 'ung' enden.

\$element =~ /^[p{Lu}]\p{L}*\p{V}ung\$/ oder **\$element =~ /^[p{Lu}]\p{L}*\p{V}ung\p{L}*/\$** (auch z.B. Umleitungen und Umleitungs)

3 .7. alle Wörter die nur aus Großbuchstaben bestehen und maximal 8 Buchstaben lang sind.

\$element =~ /^[p{Lu}]{1,8}\$/

3 .8. alle Wörtern in denen kein Vokal vorkommt.

\$element =~ /^[^aeiouäöü]+\$/i oder **\$element !~ /aeiouäöü/i**

4. Schreiben Sie ein PERL Programm, das die Anzahl aller kleingeschriebenen Wörter in der Datei wittgenstein.txt ermittelt. Geben Sie die Anzahl aus.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: liest die Anzahl alle kleingeschriebenen Wörter aus
use utf8;
use locale;
use strict;
{
    my ($zeile, %lexikon, @woerter, $element);
    my $zaehler = 0;
    open(ARTIKEL, "<utf8", "wittgenstein.txt") or die "File not found!";

    while ($zeile = <ARTIKEL>) {
        chomp($zeile);
        @woerter = split(/[\p{Z}\p{P}]+/, $zeile);

        foreach $element (@woerter) {
            $lexikon{$element]++;
        }
    }

    foreach $element (sort {$lexikon{$a} <=> $lexikon{$b}} keys %lexikon) {
        if ($element =~ /^[^p{LI}]$/) {
            $zaehler++;
        }
    }

    print "Anzahl kleingeschriebener Wörter: $zaehler\n";
    close ARTIKEL;
}
```

5. Geben Sie alle Zeilen der Datei aus, die mit einem Grossbuchstaben beginnen.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: gibt alle Zeilen aus die mit Grossbuchstaben beginnen
use utf8;
use locale;
use strict;
{
    my ($zeile);
    open(ARTIKEL, "<:utf8", "wittgenstein.txt") or die "File not found!";

    while ($zeile = <ARTIKEL>) {
        chomp($zeile);
        if($zeile =~ /\p{Lu}/) {
            print"$zeile\n";
        }
    }

    close ARTIKEL;
}
```

6. Geben Sie alle Zeilen der Datei aus, in denen das Wort "der", "die" oder "das" vorkommt.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: gibt alle Zeilen aus in denen der/ die/ das vorkommt
use utf8;
use locale;
use strict;
{
    my ($zeile);
    open(ARTIKEL, "<:utf8", "wittgenstein.txt") or die "File not found!";

    while ($zeile = <ARTIKEL>) {
        chomp($zeile);
        if($zeile =~ /\b[Dd]er\b|\b[Dd]ie\b|\b[Dd]as\b/) {
            # oder:
            #if($zeile =~ /\b[Dd](er|ie|as)\b/) {
                print"$zeile\n";
            }
        }

    close ARTIKEL;
}
```

7. Geben Sie alle Wörter der Datei aus, die von einem Satzzeichen gefolgt sind!

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: gibt alle Wörter aus die von einem Satzzeichen gefolgt sind
use utf8;
use locale;
use strict;
{
    my ($zeile, %lexikon, @woerter, $element);
    open(ARTIKEL, "<:utf8", "wittgenstein.txt") or die "File not found!";

    while ($zeile = <ARTIKEL>) {
        chomp($zeile);
        @woerter = split(/ /, $zeile);
        # dafür müssen wir Satzzeichen in den Hash mitaufnehmen

        foreach $element (@woerter) {
            $lexikon{$element}++;
        }
    }

    foreach $element (sort {$lexikon{$a} <=> $lexikon{$b}} keys %lexikon) {
        if ($element =~ /\p{L}+(\p{P}|\p{Z})$/ ) {
            print "$element\n";
        }
    }

    close ARTIKEL;
}
```

8. Schreiben Sie ein Programm, das ein "Lieblingswort" einliest und nachsieht, ob das von rückwärts gelesene Lieblingswort als Teil eines Wortes in einer großen Datei vorkommt.

Beispiel: "LOVE" => "EVOL" => Revolution => "R.EVOL.ution"

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: testet ob rückwärts gelesenes Wort in Wort der Datei vorkommt
use utf8;
use locale;
use strict;
{
    my ($zeile, %lexikon, @woerter, $element, $elementNeu);
    open(ARTIKEL, "<:utf8", "sz_1Mio.txt") or die "File not found!";

    while ($zeile = <ARTIKEL>) {
        chomp($zeile);
        @woerter = split(/[\p{Z}\p{P}]+/, $zeile);

        foreach $element (@woerter) {
            $lexikon{$element]++;
        }
    }

    print"Bitte geben Sie Ihr Lieblingswort ein:>>>\n";
    my $liebling = <>;
    chomp($liebling);
    $liebling = lc($liebling);
    my $lieblingRW = reverse($liebling);
    print $lieblingRW;

    foreach $element (sort {$lexikon{$a} <=> $lexikon{$b}} keys %lexikon) {
        if ($element =~ /($lieblingRW)/) {
            $elementNeu = uc($lieblingRW);
            $elementNeu = ".$elementNeu.";
            $element =~ s/$1/$elementNeu/;
            print "$element\n";
        }
    }

    close ARTIKEL;
}
```