Klausur zur Vorlesung
Software Engineering Praxis
Perl
8. Februar 2001

Bearbeitungszeit: 60 Min.

Name:	
Vorname:	
Matrikelnummer:	
Studiengang:	

Aufgabe	Punkte	Bewertung
1	**10**	
2	**10**	
3	**10**	
4	**10**	
Summe:	**40**	

Bitte benutzen Sie für die Lösungen den freigelassenen Platz nach der jeweiligen Angabe; sollte dieser nicht ausreichen, verwenden Sie bitte die Rückseite!
Viel Erfolg!!!

1 Datenstrukturen (10 Punkte)

Auf Ihrem Computer gibt es ein Programm `addresses`, das Ihnen Personendaten aus einem Adressbuch liefert. Jeder Datensatz besteht immer aus einer Zeile, die einzelnen Felder sind durch Doppelpunkt getrennt. Im dritten Feld stehen die Vornamen der Kinder (Trenner ist Whitespace), im vierten der vollständige Name des Vaters der Kinder (Vornamen Nachname, Trenner wieder Whitespace). Bestimmen Sie die fünf häufigsten Vornamen. Geben Sie diese nach Häufigkeit sortiert aus. Da es viele Adressen sind, schreiben Sie natürlich ein kurzes Perl-Skript. Es darf angenommen werden, dass beide relevanten Felder immer Daten enthalten.

```
blubber:suelz:Peter Paul Mary:Karl Otto Maier:Daten:...\n

mehr:xyz:Otto:Franz Peter Schmidt:blubb\n
```

```
#!/usr/local/bin/perl -w

open(FH, "addresses |") or die "Cannot fork addresses\n";
while(<FH>) {
   @first = split / /, (split /:/)[2..3];
   foreach $first (@first[0..$#first-1]) {
      $first{$first}++;
   }
}
close FH;
foreach $name
   ((sort{$first{$b} <=> $first{$a}}(keys %first))[0..4]) {
   print "$name: $first{$name}\n";
}
```

2 Filter (10 Punkte)

Schreiben Sie einen Filter in Perl, der alle C Kommentare löscht. Diese beginnen mit `/*` und enden auf `*/`. Diese können über mehrere Zeilen gehen. Geschachtelte Kommentare enden auf auf die erste Folge von `*/`.

```
#!/usr/local/bin/perl -w

($_ = join('', <>)) =~ s{/\*.*?\*/}{}sg;
print;
```

3 McCabe (10 Punkte)

Eine gute Näherung für die McCabe'sche Kennzahl eines C-Programms ist die Anzahl der `if`, `while` und `do` Schlüsselworte. Schreiben Sie eine Perl-Prozedur, die diese Kennzahl so bestimmt. Als Parameter sollte sie eine Liste von Dateinamen erhalten und einen Zeiger auf eine Liste von Kennzahlen zurückliefern. Sie dürfen annehmen, dass die C-Programme bereits durch Aufgabe 2 bearbeitet worden sind, also keine Kommentare mehr enthalten.

```
#!/usr/local/bin/perl -w

sub mccabe {
   my @res = ();
   foreach my $filename (@_) {
      my $mccabe = 0;
      open(FH, $filename) || die "Cannot open $filename\n";
      while (<FH>) {
 $mccabe += s/\b(if|while|do)\b//g;
      }
      close FH;
      push @res, $mccabe;
   }
   return \@res;
}

my $result = mccabe(@ARGV);
print join(" ", @$result), "\n";
```

4 CGI (10 Punkte)

Auf dem Computer, auf dem Ihr Webserver läuft, gibt es unter /usr/bin das Programm id. Es bekommt als Parameter einen Login-namen:

```
thales$ id melzer
uid=502(melzer) gid=200(sai)
```

Schreiben Sie in Perl ein CGI-Skript, das nach Eingabe eines Logins die User- und die GruppenID des Benutzers liefert. Versuchen Sie Ihr Skript zum Wohle Ihres Servers gegen Missbrauch zu schützen.

```
#!/usr/local/bin/perl -Tw

use strict;
use CGI;

$ENV{PATH}= "/usr/bin";
my $q = new CGI;
print $q->header, $q->start_html("id"), $q->h1("id");
if ($q->param()) {
   my $login = $q->param("login");
   if (defined($login) && $login =~ /^(\w{1,8})$/) {
      `/usr/bin/id "$1"` =~ /uid=(\d+).*gid=(\d+)/; # alt. open
      print $q->p("id of $login: UID: $1, GID: $2");
   } else {
      print $q->p("Do not try to fool me");
   }
   print $q->hr();
}
print $q->start_form(), $q->p("Login: ", $q->textfield("login"));
print $q->end_form(), $q->end_html;
```