

Teil 1

| | |
|--|--|
| Vergleichen Sie den vorliegendem Quellcode Q1, Q2 und Q3. Geben Sie die wesentlichen Unterschiede an. Geben Sie an, welcher Quellcode sinnvoller ist. Begründen Sie! | |
| Q1 | <pre>#!/usr/bin/perl -w \$DNA_Sequenz = 'ACGGTGCTGGACTGATTAGCGTATATAGC'; print \$DNA_Sequenz; print "\n"; exit;</pre> |
| Q2 | <pre>#!/usr/bin/perl -w print "\nBitte Sequenz eingeben und Zeilenvorschub druecken: "; \$DNA_Sequenz = <STDIN>; chomp \$DNA_Sequenz; print "Die eingebene Sequenz lautet: \$DNA_Sequenz"; print "\n"; exit;</pre> |
| Q3 | <pre>#!/usr/bin/perl -w print "\nBitte Sequenz eingeben und Zeilenvorschub druecken: "; \$DNA_Sequenz = <STDIN>; chomp \$DNA_Sequenz; print "Die eingebene Sequenz lautet: \$DNA_Sequenz\n"; exit;</pre> |
| → | |

Teil 2

Finden Sie passende Kommentarzeilen für Q4 und ergänzen Sie diese!
Vergleichen Sie Q4 und Q5!

| | |
|----|--|
| Q4 | <pre>#!/usr/bin/perl -w print "\nBitte Sequenz eingeben und Zeilenvorschub druecken: "; \$DNA_Sequenz = <STDIN>; chomp \$DNA_Sequenz; print "Die eingebene Sequenz lautet: \$DNA_Sequenz"; print "\n"; exit;</pre> |
| Q5 | <pre>#!/usr/bin/perl -w print "Bitte Sequenz eingeben und Zeilenvorschub druecken: "; \$DNA_Sequenz = <STDIN>; chomp \$DNA_Sequenz; print "Die eingebene Sequenz lautet: \$DNA_Sequenz"; print "\n"; exit;</pre> |
| → | |

Teil 3

Die Codezeile 2 in Q5 scheint, wie Q6 zeigt, nicht unbedingt erforderlich.
Warum ist sie dennoch sinnvoll?

| | |
|----|--|
| Q5 | <pre>#!/usr/bin/perl -w print "Bitte Sequenz eingeben und Zeilenvorschub druecken: "; \$DNA_Sequenz = <STDIN>; chomp \$DNA_Sequenz; print "Die eingebene Sequenz lautet: \$DNA_Sequenz"; print "\n"; exit;</pre> |
| Q6 | <pre>#!/usr/bin/perl -w # print "Bitte Sequenz eingeben und Zeilenvorschub druecken: "; \$DNA_Sequenz = <STDIN>; chomp \$DNA_Sequenz; print "Die eingebene Sequenz lautet: \$DNA_Sequenz"; print "\n"; exit;</pre> |
| → | |

Teil 4

| | |
|--|--|
| <p>Die Skripte Q7 und Q8 unterscheiden sich nur minimal (siehe chomp-Befehlszeile). Setzen Sie in beiden Skripten, Q7 und Q8, in der Zeile, die dem chomp-Befehl folgt, einen Breakpoint. (Das gelingt Ihnen mit Hilfe des DzSoft Perl Editors oder eines anderen Perl Editors.) Ein Breakpoint bewirkt, dass das Programm nur bis zur gewünschten Programmzeile abläuft und nicht weiter. Werte, die bis zu diesem Zeitpunkt in Variablen gespeichert werden, werden dann auf dem Bildschirm angezeigt. Beachten Sie, wie sich der Variablenwert unterscheidet, wenn Sie in beiden Fällen, Q7 und Q8, jeweils die gleiche Zeichenfolge für die Variable eingeben. Finden Sie eine passende Erklärung.</p> | |
| Q7 | <pre>#!/usr/bin/perl -w print "Bitte Sequenz eingeben und Zeilenvorschub druecken: "; \$DNA_Sequenz = <STDIN>; chomp \$DNA_Sequenz; print "Die eingebene Sequenz lautet: \$DNA_Sequenz"; print "\n"; exit;</pre> |
| → | Variablenwert aus Q7 bei gesetztem Breakpoint: |
| Q8 | <pre>#!/usr/bin/perl -w print "Bitte Sequenz eingeben und Zeilenvorschub druecken: "; \$DNA_Sequenz = <STDIN>; # chomp \$DNA_Sequenz; print "Die eingebene Sequenz lautet: \$DNA_Sequenz"; print "\n"; exit;</pre> |
| → | Variablenwert aus Q8 bei gesetztem Breakpoint: |
| → | Und die Moral von der Geschichte ... |