

Aufgabe 1: Sortierverfahren

Sortieren Sie die folgende Liste von Zahlen aufsteigend mit dem Verfahren *Insertion Sort*:

38 18 5 21 29 14 35

Geben Sie die Liste nach jedem Durchlauf der inneren Schleife an, d.h. nach jedem vollständigen Einsortieren eines Elements.

38 | 18 5 21 29 14 35
18 38 | 5 21 29 14 35
5 18 38 | 21 29 14 35
5 18 21 38 | 29 14 35
5 18 21 29 38 | 14 35
5 14 18 21 29 38 | 35
5 14 18 21 29 35 38 |

Aufgabe 2: Syntax von Programmen

Welche der folgenden Variablen Deklarationen sind syntaktisch korrekt? Bei den nicht korrekten Zeilen erläutern Sie jeweils kurz den Fehler.

```
1  x := 42
2  var int y 55 ← Typ und Variable vertauscht '=' fehlt
3  int z = 42 ← var fehlt / Typ und Variable vertauscht
4  s := string([]byte{'a', 'b', 'c'})
5  b := []byte{'a', 'b', 'c'}
6  var l1 []int := make([]int, 0) ← var und := nicht gemeinsam verwendbar
7  string := "hallo"
```

Aufgabe 3: Fehlersuche: Syntaxfehler

Das folgende Programm enthält eine Reihe an Syntaxfehlern, durch die es nicht compiliert. Markieren Sie alle Zeilen, die einen Fehler enthalten und erläutern Sie kurz, was jeweils falsch ist.

```
1 package main
2
3 import "fmt"
4
5 func PrintSomething(what string) string {
6     fmt.Println(what)
7     fmt.Println("\n")
8 }
9
10 func ComputeProduct(numbers int) int {
11     result := 1
12     for _, num := range numbers { ← num int kann nicht iterieren
13         result *= num
14     }
15     return result
16 }
17
18 func main() {
19     p = ComputeProduct(1, 3, 5, 2, 0, 2)
20     PrintSomething(string fmt.Sprint(p))
21 }
```

fehlender return

fehlender :

↑ Typ darf nicht angegeben werden

Hinweis: Es geht hier nicht um inhaltliche Fehler, nur um Syntaxfehler.

Anmerkung: Für jede falsch markierte Zeile gibt es Punktabzug!

Aufgabe 4: Fehlersuche: Inhaltliche Fehler

Die folgende Funktion ist zwar syntaktisch korrekt, sie erfüllt aber nicht ihre Aufgabe. Erläutern Sie den/die Fehler und machen Sie einen Vorschlag zur Korrektur.

```
1 // IsPrime liefert true, falls n eine Primzahl ist.
2 func IsPrime(n int) bool {
3     for i := 2; i < n-1; i++ {
4         if n%i == 0 {
5             return false
6         } else {
7             return true
8         }
9     }
10    return true
11 }
```

kann weggelassen werden

Anmerkung: Ihr Korrekturvorschlag muss kein syntaktisch korrekter Code sein. Eine Erklärung in Worten genügt.

zwischen Zeile 2 und 3

```
if n <= 1 {
    return false
}
```

Aufgabe 5: Programmverständnis

Erläutern Sie, was die folgende Funktion berechnet. Geben Sie eine möglichst allgemeine bzw. abstrakte Erklärung an. Erklären Sie auch, mit welcher Art von Argumenten diese Funktion sinnvoll arbeitet.

```
1 func Foo(m, n int) bool {  
2     if m == 0 || m > n {  
3         return false  
4     }  
5     if n == m {  
6         return true  
7     }  
8     return Foo(m, n-m)  
9 }
```

berechnet, ob m ein Teiler von n ist

Wenn $m=0$ oder $m>n$, kann m kein Teiler von n sein

Wenn $m=n$ ist m ein Teiler

es wird so lange $\text{Foo}(m, n-m)$ returniert, bis $m >$ oder $= n$ ist

Aufgabe 6: Datenstrukturen

Entwerfen Sie eine Datenstruktur, die geeignet ist um ein Textdokument zu verwalten, wie es z.B. in Textverarbeitungen wie Microsoft Word vorkommt.

Die Struktur soll Zugriff auf folgende Metadaten bzw. Dokumentteile bieten:

- Den gesamten Text.
- Die Anzahl der Wörter im Text.
- Für jede Stelle im Text die Schriftart.
- Datum der letzten Änderung.
- Autor der letzten Änderung.
- Alle am Dokument beteiligten Autoren

Begründen Sie die Wahl Ihrer Datentypen.

Anmerkung: Sie können auch weitere Hilfs-Datentypen oder Methoden definieren. Es ist nicht notwendig, syntaktisch korrekten Code zu schreiben.

type Date struct {
 Day, Month, Year int }

type Document struct {
 Text string
 Wörteranzahl int
 Lastautor string
 LastEdit Date
 FontInfo []Fontinfo
 Authors []string }

type Fontinfo struct {
 Begin, End int
 Fontname string }