

# Kurztest WWI218 Programmieren II - Algorithmen

Name:

## Allgemeine Informationen

Alle Aufgaben ergeben insgesamt **25 Punkte**. Maximal zu erreichen sind **20 Punkte**. Somit müssen nicht alle Aufgaben bearbeitet werden um die volle Punktzahl zu erreichen. Die Wahl der zu bearbeitenden Aufgaben steht Ihnen frei. Jedoch wird empfohlen, die Aufgaben 1-3 sowie eine der Aufgaben 4.1 oder 4.2 zu bearbeiten.

Die Bearbeitungszeit beträgt **30 Minuten**. Die Bearbeitung der Aufgaben erfolgt **ausschließlich** auf den ausgehändigten Aufgabenblättern. Bei Benutzung der Rückseite ist die Aufgabennummer mit zu vermerken.

Dieser Kurztest geht zu **20%** in die Gesamtbewertung der Vorlesung ein.

## Bewertung

| Aufgabe        | Erreichte Punktzahl | Maximale Punktzahl |
|----------------|---------------------|--------------------|
| Aufgabe 1      |                     | 4                  |
| Aufgabe 2      |                     | 8                  |
| Aufgabe 3      |                     | 4                  |
| Aufgabe 4.1    |                     | 4                  |
| Aufgabe 4.2    |                     | 4                  |
| Aufgabe 5      |                     | 1                  |
| <b>Gesamt:</b> |                     | 20                 |

Datum, Ort

Bewertung, Unterschrift Dozent

## Aufgabe 1 (4 Punkte)

Zur Analyse von Algorithmenlaufzeiten haben Sie die zwei Messgrößen  $\tau$  und  $O$  kennen gelernt.

- Wo liegt der Unterschied zwischen den beiden Größen?
- Warum wird zur Bestimmung der Laufzeit nicht eine Zeitmessung auf einem Computer durchgeführt?

## Aufgabe 2 (8 Punkte)

Folgend ist ein Codeausschnitt gezeigt, der für eine gegebene Zahl  $n$  den größten ganzzahligen Teiler  $t$  (wobei  $t \neq n$ ) ermittelt. Zeigen Sie, ob der Algorithmus korrekt umgesetzt ist.

- Funktioniert der Algorithmus korrekt?
- Falls der Algorithmus nicht korrekt funktioniert, nennen Sie den Grund und geben Sie an, welche Änderung notwendig ist, damit der Algorithmus korrekt funktioniert.
- Bestimmen Sie  $O$
- Stellen Sie den (ggf. korrigierten) Algorithmus als Struktogramm oder Programmablaufplan dar.

```
1 public boolean biggestDivisor(int n){
2     for(int i=n; i>0; i--){
3         if(n%i==0){
4             return i;
5         }
6     }
7     return n;
8 }
```

### Aufgabe 3 (4 Punkte)

Im folgenden Codeausschnitt wird ein Algorithmus gezeigt der verschiedene Laufzeiten haben kann. Übergeben wird ein Integer Array (`array`) und die Größe dieses Arrays(`n`).

- In welchem Fall tritt der Worst-Case und wann der Best-Case ein? (In Hinblick auf die Laufzeit)
- Bestimmen Sie  $O$

```
1 public void func(int[] array, int n){
2     for(int i=n-1;i>=0;i--){
3         if((array[i]%2)==1){
4             for(int j=0;j<n;j++){
5                 System.out.println(array[i]+array[j])
6             }
7         }else{
8             System.out.println(array[i])
9         }
10    }
11 }
```

### Aufgabe 4.1 (4 Punkte)

Analysieren Sie den folgenden Codeausschnitt. Hinweis:  $n$  beschreibt hierbei die Länge des übergebenen Arrays. Die `System.out.println()` Funktion wird als eine Operation gewertet.

- Bestimmen Sie  $\tau(n)$  und  $\tau(4)$
- Beschreiben Sie kurz die Aufgabe des Algorithmus

```
1 public void func(int[] array, int n){  
2     for(int i=0;i<n;i++){  
3         System.out.println(array[i]*i)  
4     }  
5 }
```

### Aufgabe 4.2 (4 Punkte)

Im folgenden wird ein Struktogramm für einen Algorithmus gezeigt.

- Welche Aufgabe hat der Algorithmus?
- Setzen Sie das Struktogramm in Java-Code um. Nutzen Sie dafür die unten gegebene Vorlage.

Hinweis: Die Operation „Einlesen“ wird durch das übergeben von Funktionsargumenten realisiert und das zurückgeben von Werten über das `return` statement.

```
1 public          func(          ){  
2  
3  
4  
5  
6  
7  
8 }
```

### Aufgabe 5 (1 Punkt)

Welche ist die beste IDE? (Nicht zutreffendes bitte streichen)

- Eclipse im Default Theme
- Eclipse im Dark Theme
- IntelliJ im Default Theme
- IntelliJ im Dark Theme
- vim