

## Übungsblatt 5 – Arrays & Algorithmik

---

### Aufgabe 1 – Primzahltest

Eine Primzahl ist eine Zahl, die nur durch 1 und durch sich selbst teilbar ist.

Schreiben Sie eine Methode, die testet, ob eine gegebene Zahl eine Primzahl ist. Testen Sie diese Methode mit ein paar ausgewählten Zahlen.

### Aufgabe 2 - Buchstabenhistogramm

Es soll für einen String gezählt werden, wie oft jedes Zeichen vorkommt.

Beispiel: Anna: 2 x a, 2 x n.

Schreiben Sie eine Methode, die für einen String die Zeichen zählt. Testen Sie diese Methode mit ein paar ausgewählten Strings.

### Aufgabe 3 – Element einem Array hinzufügen

Wie Sie wissen, ist die Größe eines Arrays nicht veränderlich. Manchmal ist es aber nötig, an ein Array ein Element anzuhängen.

Schreiben Sie eine Methode, die ein Array und ein Element als Parameter übergeben bekommt. Die Methode soll dem gegebenen Array das gegebene Element hinten anfügen.

Beispiel: `anfuegen([1, 2, 3], 4)` gibt das Array `[1,2, 3, 4]` zurück

Testen Sie die Methode mit geeigneten Daten.

### Aufgabe 4 – Binäre Suche

Die binäre Suche kann verwendet werden, um in vorsortierten Arrays sehr schnell ein Element zu finden.

Aus Wikipedia:

„Zuerst wird das mittlere Element des Feldes überprüft. Es kann kleiner, größer oder gleich dem gesuchten Element sein. Ist es kleiner als das gesuchte Element, muss das gesuchte Element in der hinteren Hälfte stecken, falls es sich dort überhaupt befindet. Ist es hingegen größer, muss nur in der vorderen Hälfte weitergesucht werden. Die jeweils andere Hälfte muss nicht mehr betrachtet werden. Ist es gleich dem gesuchten Element, ist die Suche beendet.“

Schreiben Sie eine Methode, die ein sortiertes Array sowie das gesuchte Element erhält und den Index dieses Elements zurückgibt. Die Methode soll -1 zurückgeben, falls das Element nicht enthalten ist.

Testen Sie die Methode mit geeigneten Daten.

## Aufgabe 5 – Verdienstrechner

Gegeben ist das Beispiel aus der Vorlesung zu Kapitel 5 – Arrays:

```
public class Verdienstrechner {  
  
    public static double verdienst(double stunden, double lohn, double faktor) {  
        return stunden <= 8.0  
            ? stunden * lohn  
            : (8.0 + faktor * (stunden - 8.0)) * lohn;  
    }  
  
    public static void main(String[] args) {  
        final double lohn = 15.0;  
        final double faktor = 1.15;  
        double stundenMo = 8.0;  
        double stundenDi = 8.0;  
        double stundenMi = 9.0;  
        double stundenDo = 9.0;  
        double stundenFr = 6.0;  
        double stundenSa = 8.0;  
  
        double gesamt =  
            verdienst(stundenMo, lohn, faktor) +  
            verdienst(stundenDi, lohn, faktor) +  
            verdienst(stundenMi, lohn, faktor) +  
            verdienst(stundenDo, lohn, faktor) +  
            verdienst(stundenFr, lohn, faktor) +  
            verdienst(stundenSa, lohn, faktor);  
        System.out.println(gesamt);  
    }  
}
```

1. Was bedeutet die return-Anweisung in der Methode verdienst(.)? Wie könnte dies anders geschrieben werden?
2. Basierend auf dem Konzept von Arrays: wandeln Sie das gegebene Programm um, so dass:
  - a. Die Stunden in einem Array abgespeichert werden.
  - b. Der Gesamtstundenlohn mittels foreach-Schleife und der Methode verdienst(.) ermittelt und in einer Variablen abgespeichert wird und danach auf der Konsole ausgegeben wird.
  - c. Erweitern Sie Ihr Programm um die Methode aus Aufgabe 3 anfüegen(.) und ändern Sie diese so ab, dass es sich dabei um double-Werte und double-Arrays handelt.
  - d. Erweitern Sie Ihr Programm um eine Methode entfernen(), welche ein double-Array um ein Element kürzt, und zwar um das Element angegeben durch durch einen Index in der Parameterliste.
3. Testen Sie Ihr Programm für verschiedene stunden-Arrays, indem Sie weitere Stunden anfügen, auch wieder Stunden an einer bestimmten Position löschen. Geben Sie die relevanten Informationen an der Konsole aus.