



## **Algorithmen und Datenstrukturen (Studiengang Wirtschaftsinformatik 3. Semester)**

### **Aufgabenstellung für das Praktikum am 19.10.2023**

#### **Vorbemerkung:**

In der heutigen Vorlesung haben Sie die Eigenschaften Determiniertheit, Determinismus, Terminierung und Korrektheit von Algorithmen kennen gelernt.

Im heutigen Praktikum werden Sie sich in Ihren Kleingruppen mit unterschiedlichen Algorithmen beschäftigen, wobei Ihre Aufgabe darin besteht, diese im Hinblick auf die oben genannten Eigenschaften zu untersuchen.

Wahrscheinlich werden Sie in Ihrer Gruppe nicht alle Aufgaben im Rahmen des Praktikums vollständig bearbeiten können, so dass Sie hieran einzeln oder in der Gruppe gerne bis zur nächsten Vorlesung weiterarbeiten können, in der ich auf die einzelnen Themen nochmal eingehen werde.

Ich wünsche Ihnen „gute Unterhaltung“ – im Sinne einer Diskussion und gemeinsamen Lösungsfindung – sowie viel Erfolg!

#### **Aufgabe 3:**

Untersuchen Sie die folgenden fünf Algorithmen, die Sie auch in Moodle als Vorgabe finden, hinsichtlich der Eigenschaften *Determiniertheit*, *Determinismus* und *Terminierung*.

##### a) Algorithmus Zero

```
private static void algoZero (int zahl)
{
    for (int i=0; i<zahl; i++)
    {
        for (int j=zahl; j>0; j--)
            System.out.println(j++);
    }
}
```

##### b) Algorithmus One

```
private static void algoOne (int zahl)
{
    for (int i=0; i<zahl; i++)
    {
        for (int j=zahl; j>0; j--)
            System.out.println(j+i);
    }
}
```

c) Algorithmus Two

```
private static int algoTwo (int zahl1, int zahl2)
{
    int erg=0;

    if (zahl2!=0)
    {
        erg=zahl1%zahl2;
    }
    else
    {
        erg=-1;
    }
    return erg;
}
```

d) Algorithmus Three

```
private static int algoThree ()
{
    int zahl=-10+(int)(Math.random()*31);

    while (zahl<=10)
    {
        zahl*=zahl;
    }
    return zahl;
}
```

e) Algorithmus Four

```
private static float algoFour (float zahl)
{
    float zufall;

    if (zahl>10)
    {
        zufall=(float)Math.random();
        if (zahl>20)
        {
            zahl-=zufall;
        }
        zahl+=zufall;
    }

    while (zahl<2)
    {
        zahl=(float)Math.pow(zahl, 3);
    }
    return zahl;
}
```

**Aufgabe 4:**

Eine weitere Eigenschaft von Algorithmen ist die *Korrektheit*.

In Moodle finden Sie als Ausgangslage für diese Aufgabe eine Vorgabe mit folgender Funktion, die als sogenannte „Ägyptische Multiplikation“ bekannt ist:

```
private static int aegyptischeMultiplikation (int zahl1, int zahl2)
{
    int akkumulator=0;

    while (zahl2>1)
    {
        if (zahl2%2!=0)
        {
            akkumulator += zahl1;
        }
        zahl1=zahl1*2;
        zahl2=zahl2/2;
    }
    return akkumulator+zahl1;
}
```

**a) Korrektheit**

Untersuchen Sie, ob dieser Algorithmus korrekt arbeitet, wenn folgende Bedingung gilt:

$zahl1 \in \mathbb{N}$  und  $zahl2 \in \mathbb{N}$

Kann man dies mit überschaubarem Aufwand formal beweisen? Recherchieren Sie zu dieser Frage auch gerne im Internet!

Was können Sie feststellen, wenn sich die Bedingung wie folgt ändert:

$zahl1 \in \mathbb{Z}$  und  $zahl2 \in \mathbb{Z}$ ?

Können Sie das Problem beheben?

**b) Terminierung**

Wieso ist sichergestellt, dass unter der Bedingung  $zahl1 \in \mathbb{N}$  und  $zahl2 \in \mathbb{N}$  der obige Algorithmus in jedem Fall terminiert?

Viel Spaß und Erfolg bei der Bearbeitung der Aufgaben!