

Matrikelnummer:

Punkte:

Lösung der Klausur zur Vorlesung
Grundlagen der Informatik und Numerik
Dr. Monika Meiler

Bemerkungen:

- **Jedes Blatt ist mit der Matrikelnummer zu versehen.**
- **Jede Aufgabe ist auf dem vorgesehenen Blatt zu lösen. Reicht der dortige Platz nicht aus, so verwenden Sie ein mit der Matrikelnummer versehenes zusätzliches Blatt.**
- **Es sind außer Papier und Schreibzeug *keine* weiteren Hilfsmittel erlaubt (keine Taschenrechner, keine Unterlagen, . . .).**
- **Es ist leserlich und *nicht* mit Bleistift zu schreiben.**
- **Beantworten Sie Fragen pro Pfeil „>“ mit *genau* einem Sachverhalt.**

Schalten Sie Ihr Handy aus!

Maximum: 66 Punkte

Note 1: 60 Punkte Note 2: 50 Punkte Note 3: 40 Punkte Note 4: 30 Punkte

Notizen

Matrikelnummer:

Punkte:

Klausuraufgabe 1

(22 Punkte)

zu (a) Eine EDV-Anlage arbeitet nach dem EVA-Prinzip. Erklären Sie das Prinzip jeweils in einem Satz. **2P**

➤ **E ... Eingabe von Informationen, die verarbeitet werden sollen**

➤ **V ... Verarbeitung dieser durch ein Programm des Computers**

➤ **A ... Ausgabe der Ergebnisse, die verarbeiteten Informationen**

zu (b) Welche Beziehung besteht zwischen einem Algorithmus und einem Programm? **2P**

➤ **Ein Programm ist ein codierter Algorithmus.**

zu (c) Geben Sie die Additions- und Multiplikationstabelle zum Rechnen im Positionssystem zur Basis $b = 3$ an. **4P**

$*\backslash+$	0	1	2
0	0\0	1	2
1	0	1\2	10
2	0	2	11\11

zu (d) Berechnen Sie in diesem System mit Zwischenergebnissen und machen Sie die Probe im Dezimalsystem: **8P**

$$\begin{array}{r} (122)_3 \text{ Probe: } (2+6+9) \\ + (12)_3 \quad + (3+2) \\ \hline \text{Übertrag } 110 \\ \hline \underline{\underline{(211)_3}} \quad = (1+3+18) \end{array} \quad \begin{array}{r} (122)_3 * (12)_3 \text{ Probe: } 17*5=85 \\ \hline 1220 \\ \hline 1021 \\ \hline \text{Übertrag } 11100 \\ \hline \underline{\underline{(10011)_3}} \quad = 1+3+81 \end{array}$$

zu (e) Wozu sind in einem Programm Datentypangaben notwendig?

2P

➤ *Speicherplatzbereitstellung*

➤ *Interpretationsvorschrift*

zu (f) Welche (mathematische) Größen werden in den folgenden Variablen gespeichert? 4P

Variable	(mathematische) Größe
<code>int i;</code>	<code>i</code> ist eine Variable für <i>eine ganze Zahl</i>
<code>String str;</code>	<code>str</code> ist eine Variable für <i>eine Zeichenkette</i>
<code>double[] v;</code>	<code>v</code> ist eine Variable für <i>einen Vektor von Gleitpunktzahlen</i>
<code>boolean b;</code>	<code>b</code> ist eine Variable für <i>einen Wahrheitswert</i>

Matrikelnummer:

Punkte:

Klausuraufgabe 2

(22 Punkte)

Berechnen Sie schrittweise mittels der Trapezregel und mittels der Simpsonregel das Flächenintegral $F = \int_a^b f(x)dx = \int_{-1}^3 ((1-x)^2 - 4)dx$: $F = \int_{-1}^3 (x^2 - 2x - 3)dx$

zu (a) Unterteilen Sie das Intervall $[-1,3]$ in 4 Teilintervalle. Berechnen Sie die Funktionswerte an den 5 Stützstellen. Verwenden Sie für die Berechnung von $f(2)$ und $f(3)$ das Horner Schema. **7P**

$$f(-1) = 0$$

$$f(0) = -3$$

$$f(1) = -4$$

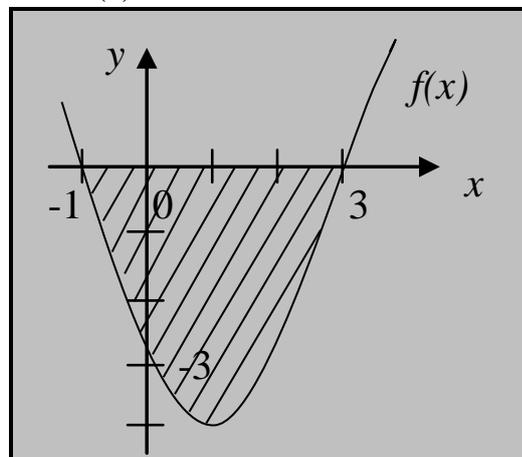
$$f(2) = -3$$

1	-2	-3	
0	2	0	2
<hr/>			
1	0	-3	

$$f(3) = 0$$

1	-2	-3	
0	3	3	3
<hr/>			
1	1	0	

zu (b) Skizzieren Sie die Funktion $f(x)$ und markieren Sie die zu berechnende Fläche. **3P**



zu (c) Berechnen Sie die Fläche F mittels der Trapezregel und mittels der Simpsonregel. **6P**

➤ Trapezregel: $F \approx \frac{1}{2}(0 - 6 - 8 - 6 + 0) = -10$

➤ Simpsonregel: $F \approx \frac{1}{3}(0 - 12 - 8 - 12 + 0) = -\frac{32}{3} = -10\frac{2}{3}$

zu (d) Berechnen Sie den mathematischen Wert. **2P**

➤ Mathematischer Wert:

$$F = \int_{-1}^3 ((1-x)^2 - 4) dx = \left[\frac{1}{3}x^3 - x^2 - 3x \right]_{-1}^3 = (9 - 9 - 9) - \left(-\frac{1}{3} - 1 + 3\right) = -\frac{32}{3} = -10\frac{2}{3}$$

zu (e) Vergleichen Sie die Endergebnisse mit dem mathematischen Wert und beantworten Sie für beide Verfahren die folgende Frage: Wie viele Stützstellen sind für dieses Beispiel notwendig, um eine korrekte Lösung zu erhalten? Begründen Sie Ihre Vermutung. **4P**

➤ Trapezregel: *Nie genau berechenbar, da Anzahl der Stützstellen stets abzählbar und die Menge der reellen Zahlen des Wertevorrates nicht abzählbar ist.*

➤ Simpsonregel: *Drei Stützstellen reichen aus, da eine Parabel mit drei Stützstellen eindeutig bestimmt ist, d.h. die Approximation der Funktion $f(x)$ ist mit sich selbst identisch.*

Klausuraufgabe 3

(22 Punkte)

Die Menge aller rationalen Zahlen Q lassen sich als Menge aller gemeinen (gewöhnlichen) Brüche $Q = \left\{ \frac{p}{q} \mid p, q \in \mathbb{Z} \wedge q \neq 0 \right\}$ darstellen. Gegeben sei das folgende Klassendiagramm für solch eine Klasse `Bruch`, bestehend aus Zähler `zaehler` und Nenner `nenner`, einer Methode `add` zur Addition eines Bruches mit einem anderen und einer Methode `kuerzen` zum Kürzen eines Bruches. Die Klasse soll in Java implementiert werden.

Bruch	
- zaehler:	int
- nenner:	int
+ setBruch(int, int):	boolean
+ toString():	String
+ add(Bruch):	Bruch
+ getZaehler():	int
+ getNenner():	int
- kuerzen():	void

zu (a) Nehmen Sie ein Attribut für den Zaehler `zaehler` und ein Attribut für den Nenner `nenner` in die Klasse **Bruch** auf **2P**

Bruch.java

```
public class Bruch
{
    /**
     * Zaehler eines Bruchs.
     */
    private int zaehler = 0;

    /**
     * Nenner eines Bruchs.
     */
    private int nenner = 1;
}
```

zu (b) Schreiben Sie eine Methode `toString` so, dass ein Bruch $\frac{p}{q}$ in der linearen Form p/q zurückgegeben wird. **2P**

```
/**
 * Darstellung eines Bruchs.
 * @return Bruch in linearer Schreibweise
 */
public String toString()
{
    return "" + zaehler + " / " + nenner;
}
```

zu (c) Gegeben sei die folgende Methode `setBruch` mit `z` als Zähler und `n` als Nenner eines Bruchs. Tragen Sie einen javadoc-Kommentar für diese Methode ein. **2P**

```
/**
 * Setzt Zaehler und Nenner eines Bruchs.
 * @param z Zaehler des Bruchs
 * @param n Nenner des Bruchs
 * @return true, falls Bruch gesetzt, sonst false
 */
public boolean setBruch( int z, int n)
{
    if( n == 0) return false;

    zaehler = z;
    nenner = n;
    kuerzen();
    return true;
}
```

zu (d) Gegeben sei die folgende Methode `kuerzen`. Analysieren Sie diese Methode, indem Sie alle Speicherplatzveränderungen für den Bruch $\frac{10}{6}$ protokollieren. **4 P**

```
/**
 * Kuerzen eines Bruchs.
 */
private void kuerzen()
{
    int m = Math.abs( zaehler);
    int n = Math.abs( nenner);

    int r;
    do
    {
        r = m % n;
        m = n;
        n = r;
    } while( n != 0);

    if( m > 1)
    {
        zaehler /= m;
        nenner /= m;
    }
}
```

Protokoll:

zaehler	nenner	m	n	r
10	6	10	6	4
		6	4	2
		4	2	0
		2	0	
5	3			

zu (e) Programmieren Sie die Methode `add` für die Addition eines Bruchs mit einem anderen.

4P

```
/**
 * Addition.
 * @param a Bruch als zweiter Operand
 * @return Summe des Bruchs mit dem Bruch a
 */
public Bruch add( Bruch a)
{
    int z = zaehler * a.getNenner() +
        nenner * a.getZaehler();
    int n = nenner * a.getNenner();

    Bruch b = new Bruch();
    b.setBruch( z, n);

    return b;
}
```

zu (f) In einem Testprogramm seien vier Brüche $a, b, c, d \in \mathbb{Q}$ wie folgt vereinbart:

4P

```
public static void main( String[] args)
{
    Bruch a = new Bruch();
    Bruch b = new Bruch();
    Bruch c = new Bruch();
    Bruch d = new Bruch();

    a.setBruch( 2, 3);
    b.setBruch( 3, 18);
}
```

Geben Sie je eine Java-Anweisung zur Berechnung von $c = a + b$ und $d = 2 * c$ an.

```
// Beispiel c = a + b
c = a.add( b);
// Beispiel d = 2 * c
d = c.add( c);
```

zu (g) Geben Sie als Kommentare die Konsolenausgaben für c und d unter Verwendung von (b) an.

4P

```
System.out.println( "c = " + c); // c = 5 / 6

System.out.println( "d = " + d); // d = 5 / 3

}
}
```