

## Lösung der Übungsaufgabenserie 14 Grundlagen der Informatik und Numerik

---

### 1. *WurzelBerechner.java*

zu (a)  $x = \sqrt[3]{3} \Rightarrow x^2 = 3 \Rightarrow f(x) = x^2 - 3 \Rightarrow f'(x) = 2x \Rightarrow x_1 = x_0 - \frac{x_0^2 - 3}{2x_0}$ :

$$x_0 = 3 \Rightarrow x_1 = 3 - \frac{9-3}{2 \cdot 3} = 2 \Rightarrow x_2 = 2 - \frac{4-3}{2 \cdot 2} = \frac{7}{4} = 1,75$$

$$\Rightarrow x_3 = \frac{7}{4} - \frac{\frac{49}{16} - 3}{2 \cdot \frac{7}{4}} = \frac{7}{4} - \frac{\frac{1}{16}}{\frac{7}{2}} = \frac{7}{4} - \frac{1}{56} = \frac{97}{56} \approx 1,73214\dots$$

TR: 1,7320508075688772935274463415059

zu (b)

### *WurzelBerechner.java*

```
// WurzelBerechner.java
import Tools.IO.*;

/**
 * Berechnet Wurzeln.
 */
public class WurzelBerechner
{
    /**
     * Wurzelberechner:
     * Wurzeleingabe, Wertberechnung, Ergebnisausgabe.
     */
    public static void main( String[] args)
    {
        // Ueberschrift
        System.out.println();
        System.out.println( "Wurzelberechner");

        // Dialog
        char weiter = 'j';
        do
        {
            // Neue Wurzel
            System.out.println();
            Wurzel wurzel = new Wurzel();

            // Eingabe des Wurzelexponenten
            System.out.println( "Wurzeleingabe");
            if( wurzel.konsolenEingabe())
            {
                // Wurzelausgabe
                System.out.println();
                System.out.println( "f( x) = " + wurzel);

                do
                {
                    // Argumenteingabe
                    System.out.println();
                    System.out.println( "Wertberechnung");
                    double x = IOTools.readDouble
                        ( "Wurzelradikant( x > 0) x = ");
                    System.out.println( " " + x);
                } while( weiter != 'q');
            }
        } while( weiter != 'q');
    }
}
```

MM 2013

// Eingaben

```
// Wertberechnung und Ausgabe
    int k = wurzel.getExponent();
    System.out.println( k + ". Wurzel( " + x + " )");
    System.out.println
    ( "\twurzel.wert    \t= " + wurzel.wert( x));
    System.out.println
    ( "\tMath.pow      \t= " + Math.pow( x, 1./k));

// Weiter
    System.out.println();
    weiter = IOTools.readChar( "Neues Argument (j/n)? ");
    } while( weiter == 'j');
}

    weiter = IOTools.readChar( "Neuer Exponent (j/n)? ");
    } while( weiter == 'j');

// Programm beendet
    System.out.println();
    System.out.println( "Programm beendet");
}
}
```

x	$\sqrt[3]{3}$	$\sqrt[2]{66.4225}$	$\sqrt[2]{1.96}$	$\sqrt[3]{541.343375}$	$\sqrt[3]{1.9487171}$
Wurzel	1.7320508075688772	8.15	1.4	8.15	1.1
Math	1.7320508075688772	8.15	1.4	8.149999999999999	1.1

**WurzelBerechner.out**

Wurzelberechner

Wurzeleingabe  
f( x) = 2. Wurzel( x)

Wertberechnung  
2. Wurzel von 3.0  
wurzel.wert = 1.7320508075688772  
Math.pow = 1.7320508075688772

2. Wurzel von 66.4225  
wurzel.wert = 8.15  
Math.pow = 8.15

2. Wurzel von 1.96  
wurzel.wert = 1.4  
Math.pow = 1.4

2. Wurzel von -1.0  
wurzel.wert = NaN  
Math.pow = NaN

Wurzeleingabe  
f( x) = 3. Wurzel( x)

Wertberechnung  
3. Wurzel von 541.343375  
wurzel.wert = 8.15  
Math.pow = 8.149999999999999

Wurzeleingabe

```
f( x) = 7. Wurzel( x)

7. Wurzel von 1.9487171
  wurzel.wert      = 1.1
  Math.pow         = 1.1
```

Programm beendet

## 2. *PolynomNullstellen.java*

zu (a)

```
Polynomintegrator
Berechne Nullstellen von:
24.0 + -2.0 x^1 + -5.0 x^2 + 1.0 x^3
```

Mathematische Loesung: 4, 3, -2

```
Anfangswert x0 = -3
Newton (5): -2.0
```

```
Anfangswert x0 = 0
Newton (11): 3.9999999999999996
```

```
Anfangswert x0 = 2
Newton (7): 3.0000000000000004
```

zu (b) Eigenes Polynom.

## 3. *Arctan.java*

zu (a)

<b>Arctan</b>	
+ wert(double):	double
+ wertErsteAbleitung(double):	double
+ toString():	String
- myArctan(double):	double
- myArctanBesser(double):	double

zu (b)

zu (c)

### *Arctan.java*

```
// Arctan.java MM 2013

/**
 * Logarithmusfunktion f(x) = arctan( x),
 * D = R, W = ] -PI/2, PI/2[.
 */
public class Arctan extends DifferenzierbareFunktion
{
  /* ----- */
  /**
   * Berechnen eines Funktionswertes.
   * @param arg Argument
   * @return arctan( arg)
   */
  public double wert( double arg)
  {
    // Trivialfall
    if( arg == 0) return 0;
  }
}
```

```

// einfache Newtoniteration
// return myArctan( arg);
// verbesserte Newtoniteration
    return myArctanBesser( arg);
}

/**
 * Berechnen eines Funktionswertes als Nullstelle der
 * Umkehrfunktion  $\tan(x) - \text{arg}$  mittels Newton.
 * @param arg Argument
 * @return arctan( arg)
 */
private double myArctan( double arg)
{
    Iteration it = new Iteration();
    Tangens tan = new Tangens();
    return it.newton_1( tan, arg, 1e-15, 1000);
}

/**
 * Verbesserte Funktionswertberechnung fuer  $|x| > 1$ .
 * @param arg Argument
 * @return arctan( arg)
 */
private double myArctanBesser( double arg)
{
    int fak = 1;
    if( Math.abs( arg) > 1 )
    {
        Wurzel sqrt = new Wurzel();
        sqrt.setWurzel( 2);
        fak = 2;
        arg /= ( 1 + sqrt.wert( 1 + arg * arg));
    }

    return fak * myArctan( arg);
}

/**
 * Berechnen einer ersten Ableitung
 *  $f'(x) = 1 / ( 1 + x^2)$ .
 * @param arg Argument
 * @return f'( arg)
 */
public double wertErsteAbleitung( double arg)
{
    // Trivialfall
    if( arg == 0) return 0;

    // Sonst
    return 1 / ( 1 + arg * arg);
}

/* ----- */
/* toString-Methode */
/**
 * Darstellen der Arcustangensfunktion.
 * @return Funktion in linearer Schreibweise
 */
public String toString()
{
    return "arctan( x)";
}

```

}

zu (d)

**ArctanBerechner.java**

```
// ArctanBerechner.java
import Tools.IO.*;

/**
 * Berechnet arctan.
 */
public class ArctanBerechner
{
/**
 * Arcustangensberechner:
 * Argumenteingabe, Wertberechnung, Ergebnisausgabe.
 */
    public static void main( String[] args)
    {
// Ueberschrift
        System.out.println();
        System.out.println( "Arcustangensberechner");

// Arcustangensfunktion
        Arctan arctan = new Arctan();

// Tangensfunktion
        Tangens tan = new Tangens();

// Dialog
        char weiter = 'j';
        do
        {
// Argumenteingabe
            System.out.println();
            System.out.println( "Wertberechnung");
            double x = IOTools.readDouble( "x = ");
            System.out.println( x);

// Werteberechnung
            double y = arctan.wert( x);
            double yy = Math.atan( x);
            double xx = tan.wert( y);

// Ausgabe
            System.out.println
                ( "\tarctan.wert          = " + y);
            System.out.println
                ( "\tMath.atan          = " + yy);
            System.out.println
                ( "\ttan.wert( arctan.wert( " + x + " ))= " + xx);

// Weiter
            System.out.println();
            weiter = IOTools.readChar( "Neues Argument (j/n)? ");
        } while( weiter == 'j');

// Programm beendet
        System.out.println();
        System.out.println( "Programm beendet");
    }
}
```

MM 2013  
// Eingaben

**ArctanBerechner.in**

```

0
j
0.26794919243112270647255365849413
j
0.32491969623290632615587141221513
j
0.4142135623730950488016887242097
j
0.57735026918962576450914878050196
j
0.72654252800536088589546675748062
j
1
j
1.7320508075688772935274463415059
j
10
n
    
```

**ArctanBerechner.out**

Arcustangensberechner

Wertberechnung

```

x = 0.0
  arctan.wert = 0.0
  Math.atan = 0.0
  tan.wert( arctan.wert( 0.0)) = 0.0

x = 0.2679491924311227
  arctan.wert = 0.2617993877991494
  Math.atan = 0.2617993877991494
  tan.wert( arctan.wert( 0.2679491924311227)) = 0.2679491924311227

x = 0.32491969623290634
  arctan.wert = 0.31415926535897937
  Math.atan = 0.3141592653589793
  tan.wert( arctan.wert( 0.32491969623290634)) = 0.32491969623290645

x = 0.41421356237309503
  arctan.wert = 0.39269908169872425
  Math.atan = 0.39269908169872414
  tan.wert( arctan.wert( 0.41421356237309503)) = 0.41421356237309503

x = 0.5773502691896257
  arctan.wert = 0.5235987755982986
  Math.atan = 0.5235987755982988
  tan.wert( arctan.wert( 0.5773502691896257)) = 0.5773502691896255

x = 0.7265425280053609
  arctan.wert = 0.6283185307179586
  Math.atan = 0.6283185307179586
  tan.wert( arctan.wert( 0.7265425280053609)) = 0.726542528005361

x = 1.0
  arctan.wert = 0.7853981633974485
  Math.atan = 0.7853981633974483
  tan.wert( arctan.wert( 1.0)) = 1.0000000000000002
    
```

```

x = 1.7320508075688772
  arctan.wert                = 1.0471975511965972
  Math.atan                  = 1.0471975511965976
  tan.wert( arctan.wert( 1.7320508075688772)) = 1.7320508075688754

x = 10.0
  arctan.wert                = 1.4711276743037345
  Math.atan                  = 1.4711276743037347
  tan.wert( arctan.wert( 10.0)) = 10.000000000000002

```

Programm beendet

#### Hinweis zur Kontrolle

$x$	0	$2-\sqrt{3}$	$\sqrt{1-\frac{2}{5}\sqrt{5}}$	$\sqrt{2}-1$	$\frac{1}{3}\sqrt{3}$	$\sqrt{5-2\sqrt{5}}$	1	$\sqrt{3}$
$\arctan x$	0	$\frac{\pi}{12}$	$\frac{\pi}{10}$	$\frac{\pi}{8}$	$\frac{\pi}{6}$	$\frac{\pi}{5}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$

#### 4. *SonnenStand.java*

zu (a)

##### *SonnenStand.java*

```

// SonnenStand.java                MM 2010
import Tools.IO.*;                  // Eingaben

/**
 * Anwendung der Cotangensfunktion zur Berechnung
 * des Sonnenstandes.
 */
public class SonnenStand
{
/**
 * Eingabe der Schatttenlaenge des Stabes (100 cm)
 * Ausgabe der Sonnenstand
 * ( Winkel zwischen Sonnenstrahl und Horizont)
 */
  public static void main( String[] args)
  {
    char weiter = 'j';
    do
    {
//Eingabe der Messdaten
      System.out.println();
      double schatten
      = IOTools.readDouble( "Schatten [cm]: ");
      System.out.println( schatten);

      double winkel = 90;
      if( schatten != 0)
      {
// Berechnung des Sonnenstandes
        Arctan arctan = new Arctan();
        winkel= arctan.wert( 100 / schatten);
        winkel *= 180 / Math.PI;          // RAD -> GRD
      }

// Ausgabe der Baumhoehe
      if( winkel <= 90 && winkel > 0)
        System.out.println
        ( "Sonnenhoehe [GRD]: " + winkel);
    }
  }
}

```

```
        else // Fehler in den Messdaten
            System.out.println( "Messfehler in der Eingabe");

// Weiter
    System.out.println();
    weiter
    = IOTools.readChar( "Noch ein Sonnenstand (j/n)? ");
} while( weiter == 'j');

System.out.println();
System.out.println( "Berechnung beendet");
}
}
```

zu (b)

**SonnenStand.in**

```
224
j
100
j
81
j
119
j
247
n
```

**SonnenStand.out**

```
Schatten [cm]: 224.0
Sonnenhoehe [GRD]: 24.05734945001179

Noch ein Sonnenstand (j/n)?
Schatten [cm]: 100.0
Sonnenhoehe [GRD]: 45.000000000000014

Noch ein Sonnenstand (j/n)?
Schatten [cm]: 81.0
Sonnenhoehe [GRD]: 50.99252744787977

Noch ein Sonnenstand (j/n)?
Schatten [cm]: 119.0
Sonnenhoehe [GRD]: 40.04154908051488

Noch ein Sonnenstand (j/n)?
Schatten [cm]: 247.0
Sonnenhoehe [GRD]: 22.04097232521934

Noch ein Sonnenstand (j/n)?
Berechnung beendet
```