

Lösung Übungsblatt Nr. 3 / ALP 2 zur Abgabe 06.05.2004

Aufgabe 1.

Quellcode:

```
public static float aufgabe1(float x, int e) {
    float sum=0.0f;          // das Gesamtergebnis
    float temp=1.0f;        // der aktuell errechnete Wert
    int i=1;                // Zählervariable

    do {
        temp=pot(x,i)/(float)fak(i);          // aktuellen Wert zusammenbauen
        if ((--i%4)==0) sum+=temp; else sum-=temp; // entscheiden ob Gesamtwert + oder - aktuellem Wert
        i+=2;                                // Zählervariable um 2 erhöhen
    } while (((temp/sum)*100)>e);             // „do“ solange, wie aktuell errechneter Wert größer als e % vom Gesamtwert ist.
    return sum;
}

public static int fak(int i) {
    if (i==1) return 1;
    else return (i*fak(i-1));
}

public static float pot(float x, int i) {
    float sum=1.0f;
    while (i>0) { sum*=x; i--; }
    return sum;
}
```

Kommentierung:

x ist die eigentliche Eingabe-Variable und e bestimmt den Prozent-Wert, ab dem die Potenzreihe abgebrochen wird.

Die Methode berechnet nach der gegebenen Formel einen Wert. Da bei mir die Ergebnisse der Formel nicht mit der sinus-funktion übereinstimmen, wohl aber die Programmausgabe und der Taschenrechner gehe ich von einem Fehler in der Formel aus.

Testlauf:

```
x=0.5, e=1 --> 0.3515625
x=1.0, e=1 --> 0.2986111
x=1.0, e=20 --> 0.2916666
```

Lösung Übungsblatt Nr. 3 / ALP 2 zur Abgabe 06.05.2004

Aufgabe 2.

Quellcode:

```
public static int show(int number, int base) {
    int sum=0;           // sum wird das ergebnis
    int i=1;            // i ist der Stellenmultiplikator
    boolean neg=false;  // negative Eingabe true,false

    if (number<0) { neg=true; number*=-1; } // Negative Eingabe erkennen, merken und ändern
    while (number>0) { // Solange wie number größer 0
        sum+=((number%base)*i); // Ergebnis = das bisherige Ergebnis + (Stellenmultiplikator * (number modulo base))
        number/=base; // number durch base dividieren
        i*=10; // Stellenmultiplikator erhöhen
    }
    if (neg==true) sum*=-1; // Negative Eingabe an Ausgabe weiterreichen
    return sum;
}
```

Kommentierung:

Um eine Zahl aus dem Dezimalsystem in ein anderes Zahlensystem umzurechnen dividiert man sie fortlaufend durch die neue basis und notiert die Reste.

Die fortlaufende Dividierung übernimmt die Schleife.

Das $i*=10$ am Ende ist ein stellenmultiplikator, welcher dazu benutzt wird, die Reste hintereinander in eine Zahl zu schreiben.

Testlauf:

number=10, base=2 --> 1010

number=8, base=4 --> 20

number=356, base=6 --> 1352

Lösung Übungsblatt Nr. 3 / ALP 2 zur Abgabe 06.05.2004

Aufgabe 3)

Quellcode

```
public static void zahlZuText(int n) {
    boolean exept=true;          // Sonderfall aufgetreten true,false

    switch (n) {                 // Sonderfälle abgreifen
        case 0: System.out.print("null"); break;
        case 1: System.out.print("eins"); break;
        case 11: System.out.print("elf"); break;
        case 12: System.out.print("zwölf"); break;
        case 17: System.out.print("siebzehn"); break;
        default: exept=false;    // kein sonderfall aufgetreten.
    }
    if (exept==false) {         // kein sonderfall aufgetreten?
        int second=n%10;        // einerstelle
        int first=n/10;         // zehnerstelle
        switch (first) {       // wenn zehnerstelle gleich :
            case 0: { write(second); } break;
            case 1: { write(second); System.out.print("zehn"); } break;
            case 2: { write(second); if (second!=0) System.out.print("und"); System.out.print("zwanzig"); } break;
            case 3: { write(second); if (second!=0) System.out.print("und"); System.out.print("dreiig"); } break;
            case 4: { write(second); if (second!=0) System.out.print("und"); System.out.print("vierzig"); } break;
            case 5: { write(second); if (second!=0) System.out.print("und"); System.out.print("fnfzig"); } break;
            case 6: { write(second); if (second!=0) System.out.print("und"); System.out.print("sechzig"); } break;
            case 7: { write(second); if (second!=0) System.out.print("und"); System.out.print("siebzig"); } break;
            case 8: { write(second); if (second!=0) System.out.print("und"); System.out.print("achtzig"); } break;
            case 9: { write(second); if (second!=0) System.out.print("und"); System.out.print("neunzig"); } break;
            default: break;
        }
    }
}

// Hilfsfunktion „write“ auf der nchsten Seite !
```

Lösung Übungsblatt Nr. 3 / ALP 2 zur Abgabe 06.05.2004

```
public static void write(int n) { // gibt die einerstellen aus
    switch (n) { // wenn einerstelle gleich:
        case 1: System.out.print("ein"); break;
        case 2: System.out.print("zwei"); break;
        case 3: System.out.print("drei"); break;
        case 4: System.out.print("vier"); break;
        case 5: System.out.print("fünf"); break;
        case 6: System.out.print("sechs"); break;
        case 7: System.out.print("sieben"); break;
        case 8: System.out.print("acht"); break;
        case 9: System.out.print("neun"); break;
        default: break;
    }
}
```

Kommentierung erübrigt sich, siehe Quellcode.

Testlauf

23 --> dreiundzwanzig
99 --> neunundneunzig
0 --> null
17 --> siebzehn
20 --> zwanzig