

Anwendungsorientierte Programmierung II

Hinweise zur Übung 3

1. Arrays

Deklaration

```
Typ[] name=new Typ[anzahl];    //Typ – primitiver Datentyp/Objekttyp  
Typ name[]=new Typ[anzahl];    //möglich, aber nicht empfohlen
```

Arraylänge

Festlegung durch Größenangabe: `int[] werte= new int[5];`

Festlegung durch Initialisierung: `int[] werte={1,5,3,9,2};`

Die Indizierung beginnt bei 0.

Elementzugriff

```
//Zählschleife  
for(int i=0;i<werte.length;i++)  
{  
    System.out.println(werte[i]);  
}  
  
//for-each-Anweisung  
for(int w : werte)  
{  
    System.out.println(w);  
}
```

2. Klasse Arrays

Enthält Methoden für die Arbeit mit Feldern (Vergleichen, Filtern, Sortieren,...).

```
Arrays.sort(werte);    //sortiert die Arrayelemente aufsteigend
```

3. Pseudozufallszahlen

```
java.util.Random
```

Initialisierung des Zufallszahlengenerators

```
Random zufall = new Random();
```

Erzeugung von Pseudozufallszahlen vom Datentyp int

```
int n = zufall.nextInt(10);    //Zufallszahlen in [0,10)
```

Erzeugung von Pseudozufallszahlen vom Datentyp double

```
double x = zufall.nextDouble();    //gleichverteilte Zufallszahlen in [0,1)  
double y = zufall.nextGaussian();    //normalverteilte Zufallszahlen mit Mittel-  
//wert 0.0 und Standardabweichung 1.0
```

Werte anderer Bereiche können über Ausdrücke erzeugt werden:

```
x = 10+zufall.nextDouble();  
y = 5+1.3*zufall.nextGaussian();
```

4. Formatierte Ausgabe

Allgemeiner Aufbau der Anweisung

```
System.out.printf(formatString, varliste);
```

Parameter:

varliste	(Komma-)Liste der auszugebenden Variablen
formatstring	Zeichenkette, die die Formatierung dieser Variable(n) beschreibt, Formatierungssymbole sind am %-Zeichen erkennbar

Wichtigste Formatierer

%d ganze Dezimalzahl

%f Gleitkommazahl

%s String

Bei der Ausgabe werden die Werte der Variablen aus varliste in der vorgegebenen Reihenfolge anstelle der Formatierungsangaben in den Ausgabebetext eingesetzt.

Beispiele:

```
int m=4;
System.out.printf("m=%3d",m);           //Ausgabe: m=  4    rechtsbündig

int m=4;
System.out.printf("m=%-3d",m);          //Ausgabe: m=4    linksbündig

double x1=3.7;
double x2=5.4;
System.out.printf("x1=%8.3f, x2=%6.2f\n",x1,x2);
//Ausgabe: x1=  3,700, x2=  5,40
//\n erzeugt Newline
```

5. Klasse Decimal Format

```
java.text.DecimalFormat;
```

```
DecimalFormat df = new DecimalFormat("##0.00##");
double d=Math.PI;
System.out.println(d);           //Ausgabe: 3.141592653589793
System.out.println(df.format(d)); //Ausgabe: 3,1416
```

6. Berechnung des Median

Für die Bestimmung des Median muss das Feld zunächst sortiert werden. Weiterhin ist festzustellen, ob die Anzahl n der Elemente im Feld gerade ist oder nicht.

Modulo-Operator %

```
if (n%2== 0)           // → n gerade
{
    //...
}
```

Bit-Operator & (UND)

```
if ((n&1)==0)          // → n gerade
{
    //...
}
```

Binäre Bitoperatoren (**&** - UND und **|** - ODER) werden bitweise auf die Operanden angewendet. Das letzte Bit ergibt, ob n gerade oder ungerade ist:

n=7 :	n	0000 0111		
	<u>1</u>	<u>0000 0001</u>		
	n & 1	0000 0001	n & 1 == 1	→ n ist ungerade
n=8:	n	0000 1000		
	<u>1</u>	<u>0000 0001</u>		
	n & 1	0000 0000	n & 1 == 0	→ n ist gerade

7. Ausgabefunktion (für die Messwerte)

Allg. Syntax von (statischen) Funktionen in Java:

```
public static rtyp fname([parameter])
{
    //rtyp ... Rückgabetyt der Funktion
    //parameter ... Liste der Parameter
}
```

Java-Funktionen werden als Block im Quelltext eingefügt. Im Unterschied zu C++ ist kein Prototyp erforderlich, auch wenn die Funktionsdeklaration erst nach der `main()`-Methode steht.

8. Überprüfung der Ergebnisse

Zur Überprüfung der Ergebnisse kann der Quellcode für die zufällige Generierung des Feldes auskommentiert und durch das nachfolgende Testfeld ersetzt werden

```
double[] werte={7,2,3,5,1,7,8,9,4,6};
```

Wenn das Programm korrekt arbeitet, sollten folgende Ergebnisse in der Konsole erscheinen:

Messwerte:

7,00 | 2,00 | 3,00 | 5,00 | 1,00 | 7,00 | 8,00 | 9,00 | 4,00 | 6,00 |

Mittelwert: 5,20

Standardabweichung: 2,66

Messwerte, sortiert:

1,00 | 2,00 | 3,00 | 4,00 | 5,00 | 6,00 | 7,00 | 7,00 | 8,00 | 9,00 |

Median: 5,50

Differenz zum Mittelwert: 0,30

Zufällig generierte Messwerte lassen sich im Statistik-Rechner unter

<https://www.zinsen-berechnen.de/statistik-rechner.php> überprüfen.