

```
/**
 *
 * Programm Hello World
 *
 * @author Peter René Dietmüller
 * @version 1.0, 10/98
 */

/*
 * Jedes Java-Programm besteht aus einer Klasse, die eine
 * Methode main enthält. Die Methode main enthält das
 * Hauptprogramm.
 */

public class Hello {

    // -- Hauptprogramm --
    public static void main(String[] args) {
        System.out.println("Hello World\n");
    }

}
```

```
/**
 *
 * Automatische String-Konvertierung, Switch,
 * Globale und Lokale Variablen, Boolesche Ausdrücke
 *
 * @author Peter René Dietmüller
 * @version 1.0, 98/10
 *
 */

public class Hello2 {

    static int i; // globale Variable
    static int j; // globale Variable

    public static void test() {
        System.out.println("Test: i = " + i);
        i = 100;
        System.out.println("Test: i = " + i);
    }

    public static boolean doIt() {
        j++;
        return true;
    }

    public static void main(String[] args) {

        char c; // lokale Variablen
        int i; // -- " --
        long l; // -- " --
        String s; // -- " --

        System.out.println("Programmstart");

        /* -- Initialisierung -- */
        c = 'A';
        i = 500;
        l = 12000;
        s = "Hello World\n";
    }
}
```

```
/* -- Ausgabe -- */
System.out.print("c = ");
System.out.println(c);
System.out.print("i = ");
System.out.println(i);
System.out.print("l = ");
System.out.println(l);
System.out.print("s = ");
System.out.println(s);
/* -- Ausgabe (autom. String-Konv.) -- */
System.out.println("c = " + c);
System.out.println("i = " + i);
System.out.println("l = " + l);
System.out.println("s = " + s);
/* -- Inkrement, Dekrement -- */
System.out.println(i++ + ":" + i); // "500:501"
System.out.println(++i + ":" + i); // "502:502"
System.out.println(i + ":" + i++); // "502:502"
System.out.println(i + ":" + ++i); // "503:504"
System.out.println(i-- + ":" + i); // "504:503"
System.out.println(--i + ":" + i); // "502:502"
System.out.println(i + ":" + i--); // "502:502"
System.out.println(i + ":" + --i); // "501:500"
/* -- Vorsicht -- */
System.out.println(i + 1); // 12500
System.out.println(1 + 2 + "PCs"); // "3PCs"
System.out.println("PC" + 1 + 2); // "PC12"
System.out.println(i + c + 1); // 12565
System.out.println(i + ':' + 1); // 12558
System.out.println(i + ":" + 1); // "500:12000"
/* -- switch -- */
System.out.println("\nSwitch");
switch(i) {
    case 100:
        System.out.println("100");
        break;
    case 499:
    case 500:
    case 501:
        System.out.println("499 - 501");
        break;
    default:
        System.out.println("default");
        break;
}
```

```
/* -- Variablen -- */
System.out.println("\nVariablen");
System.out.println("i = " + i);
test();
System.out.println("i = " + i);

/* -- Boolesche Ausdrücke -- */
System.out.println("\nBoolesche Ausdrücke");
j = 0;
System.out.println("j = " + j);
if ((j == 0) && (doIt()))
    System.out.println("j == 0 && doIt()");
System.out.println("j = " + j);
if ((j == 0) && (doIt()))
    System.out.println("j == 0 && doIt()");
System.out.println("j = " + j);
if ((j == 0) & (doIt()))
    System.out.println("j == 0 & doIt()");
System.out.println("j = " + j);

}

}
```

**Programmstart**

```
c = A
i = 500
l = 12000
s = Hello World
```

```
c = A
i = 500
l = 12000
s = Hello World
```

```
500:501
502:502
502:502
503:504
504:503
502:502
502:502
501:500
12500
3PCs
PC12
12565
12558
500:12000
```

```
Switch
499 - 501
```

**Variablen**

```
i = 500
Test: i = 0
Test: i = 100
i = 500
```

**Boolesche Ausdrücke**

```
j = 0
j == 0 && doIt()
j = 1
j = 1
j = 2
```

```
/**
 *
 * Aufgabe:
 * Schreiben Sie ein Programm, das alle pythagoräischen
 * Tripel von 1 bis 100 berechnet. Ein pythagoräisches
 * Tripel besteht aus drei ganzen Zahlen, die
 * folgende Bedingung erfüllen:  $a^2 + b^2 = c^2$ 
 *
 * Lösung:
 * In drei ineinandergeschachtelten Schleifen werden
 * alle ganze Zahlen von 1 bis 100 für a, b und c
 * durchprobiert. Wenn dabei obige Bedingung erfüllt
 * wird, werden die drei Zahlen ausgegeben.
 *
 * @author Peter René Dietmüller
 * @version 1.0, 09/98
 */
import java.lang.Math;

public class Tripel {

    public static void main(String[] args) {

        final int FROM = 1;           // Konstante
        final int TO   = 100;        // Konstante

        System.out.println("Pythagoräische Tripel Var. 1");
        System.out.println("-----");
        for (int a = FROM; a <= TO; a++)
            for (int b = FROM; b <= TO; b++)
                for (int c = FROM; c <= TO; c++)
                    if (a * a + b * b == c * c)
                        System.out.println(a + "\t" + b + "\t" + c);
        System.out.println("\n");

        System.out.println("Pythagoräische Tripel Var. 2");
        System.out.println("-----");
        for (int a = FROM; a <= TO; a++)
            for (int b = FROM; b <= TO; b++)
                for (int c = FROM; c <= TO; c++)
                    if (a * a + b * b == c * c)
                        System.out.println(a + '\t' + b + '\t' + c);
        System.out.println("\n");
    }
}
```

```
System.out.println("Pythagoräische Tripel Var. 3");
System.out.println("-----");
for (int a = FROM; a <= TO; a++)
    for (int b = FROM; b <= TO; b++)
        for (int c = FROM; c <= TO; c++)
            if (a * a + b * b == c * c)
                System.out.println(a + b + c);
System.out.println("\n");

System.out.println("Pythagoräische Tripel Var. 4");
System.out.println("-----");
for (int a = FROM; a <= TO; a++)
    for (int b = FROM; b <= TO; b++)
        if (Math.sqrt(a * a + b * b) ==
            (double)Math.floor(Math.sqrt(a * a + b * b)))
            System.out.println(a + "\t" + b + "\t" +
                (int)Math.sqrt(a * a + b * b));
System.out.println("\n");

}

}
```

**Pythagoräische Tripel Variante 1**

-----  
3    4    5  
4    3    5  
5    12   13  
6    8    10  
(... einige Zeilen gelöscht ...)  
80   60   100  
84   13   85  
84   35   91  
96   28   100

**Pythagoräische Tripel Variante 2**

-----  
30  
30  
48  
42  
(... einige Zeilen gelöscht ...)  
258  
200  
228  
242

**Pythagoräische Tripel Variante 3**

-----  
12  
12  
30  
24  
(... einige Zeilen gelöscht ...)  
240  
182  
210  
224



**Pythagoräische Tripel Variante 4**

-----  
3    4    5  
4    3    5  
5    12   13  
6    8    10  
(... einige Zeilen gelöscht ...)  
80   60   100  
80   84   116  
84   13   85  
84   35   91  
84   63   105  
84   80   116  
88   66   110  
90   48   102  
90   56   106  
91   60   109  
92   69   115  
96   28   100  
96   40   104  
96   72   120  
99   20   101  
100 75   125

```
/**
 *
 * Aufgabe:
 * Schreiben Sie ein Programm, das die Zahlen von 1 bis
 * 100 aus gibt und zu jeder Zahl ermittelt und ausgibt,
 * ob es sich um eine Primzahl handelt.
 *
 * Lösung:
 * Eine Funktion IsPrim ermittelt ob eine bestimmte Zahl
 * eine Primzahl ist. Im Programm wird in einer Schleife
 * von 1 bis 100 diese Funktion aufgerufen und je nach
 * Ergebnis ein passender Text ausgegeben.
 * IsPrim prüft, ob eine Zahl eine Primzahl ist, indem
 * die gegebene Zahl durch alle ganzen Zahlen von 1 bis
 * zur Zahl - 1 dividiert wird. Wenn eine dieser
 * Divisionen keinen Rest ergibt, handelt es sich um
 * keine Primzahl. Die Schleife wird abgebrochen und der
 * Wert false zurückgegeben. Sonst wird true
 * zurückgegeben.
 *
 * Optimierung:
 * Es genügt, die Schleife bis zur Wurzel laufen zu
 * lassen.
 *
 * @author Peter René Dietmüller
 * @version 1.0, 1998/09/24
 *
 **/
```

```
public static class Prim {

    public static boolean IsPrim(int x) {

        int i;

        /* -- Prüfe, ob x durch eine ganze Zahl
           von 2 .. x-1 ohne Rest dividierbar ist -- */
        i = 2;
        while ((i < x) && (x % i != 0)) i++;    /* && <> & */

        return (i >= x);

    }

}
```

```
public static void main(String[] args) {  
  
    int i;  
  
    for (i = 1; i < 101; i++) {  
        if (IsPrim(i)) {  
            System.out.println(i + " ist eine Primzahl.");  
        } else {  
            System.out.println(i + " ist keine Primzahl.");  
        }  
    }  
  
}  
  
}
```

1 ist eine Primzahl.  
2 ist eine Primzahl.  
3 ist eine Primzahl.  
4 ist keine Primzahl.  
5 ist eine Primzahl.  
6 ist keine Primzahl.  
7 ist eine Primzahl.  
8 ist keine Primzahl.  
9 ist keine Primzahl.  
10 ist keine Primzahl.  
11 ist eine Primzahl.  
12 ist keine Primzahl.  
13 ist eine Primzahl.  
14 ist keine Primzahl.  
15 ist keine Primzahl.  
16 ist keine Primzahl.  
17 ist eine Primzahl.  
18 ist keine Primzahl.  
19 ist eine Primzahl.  
20 ist keine Primzahl.  
21 ist keine Primzahl.  
22 ist keine Primzahl.  
23 ist eine Primzahl.  
24 ist keine Primzahl.  
25 ist keine Primzahl.  
26 ist keine Primzahl.  
27 ist keine Primzahl.  
28 ist keine Primzahl.  
29 ist eine Primzahl.  
30 ist keine Primzahl.  
(... einige Zeilen gelöscht ...)  
95 ist keine Primzahl.  
96 ist keine Primzahl.  
97 ist eine Primzahl.  
98 ist keine Primzahl.  
99 ist keine Primzahl.  
100 ist keine Primzahl.

```
/**
 *
 * Aufgabe:
 * Schreiben Sie ein Programm, das von System.in zeilen-
 * weise Zahlen einliest und auf System.out ausgibt, ob
 * es sich bei der Zahl um eine Primzahl handelt.
 *
 * @author Peter René Dietmüller
 * @version 1.0, 1998/09/28
 */

import java.io.*;

public class Prim2 {

    public static boolean IsPrim(int x) {

        int i;

        /* -- Prüfe, ob x durch eine ganze Zahl
           von 2 .. x-1 ohne Rest dividierbar ist -- */
        i = 2;
        while ((i < x) && (x % i != 0)) i++;

        /* -- Wenn kein Divisior gefunden wurde,
           dann gib true zurueck. -- */
        return (i >= x);

    }

    public static void main(String[] args)
        throws java.io.IOException {

        BufferedReader in;
        String line;
        int x;

        in = new BufferedReader(
            new InputStreamReader(System.in));

        line = in.readLine(); // java.io.IOException
```

```
while (line != null) {  
    x = Integer.parseInt(line); // NumberFormatException  
    if (IsPrim(x)) {  
        System.out.println(x + " ist eine Primzahl.");  
    } else {  
        System.out.println(x + " ist keine Primzahl.");  
    }  
    line = in.readLine();  
}  
}  
}
```

**Eingabe:**

9  
3  
2  
1  
35  
5  
23  
27  
36  
100  
25  
49  
97  
113  
205  
363  
323  
123

**Ausgabe:**

9 ist keine Primzahl.  
3 ist eine Primzahl.  
2 ist eine Primzahl.  
1 ist eine Primzahl.  
35 ist keine Primzahl.  
5 ist eine Primzahl.  
23 ist eine Primzahl.  
27 ist keine Primzahl.  
36 ist keine Primzahl.  
100 ist keine Primzahl.  
25 ist keine Primzahl.  
49 ist keine Primzahl.  
97 ist eine Primzahl.  
113 ist eine Primzahl.  
205 ist keine Primzahl.  
363 ist keine Primzahl.  
323 ist keine Primzahl.  
123 ist keine Primzahl.

```
/**
 *
 * Aufgabe:
 * Schreiben Sie ein Programm, das ganze Zahlen von stdin
 * zeilenweise einliest und die Primfaktoren dieser Zahl
 * ausgibt.
 *
 * @author Peter René Dietmüller
 * @version 1.0, 1998/10
 *
 **/

import java.io.*;

public class PrimFaktoren {

    public static void PrintPrimFaktoren(int x) {

        int i;    // Laufvariable
        int p;    // Primfaktor

        System.out.print("Primfaktoren von " + x + ": ");

        p = 2;
        while (p <= x) {

            /* -- prüfen, ob p eine Primzahl ist -- */
            i = 2;
            while ((i < p) && (p % i != 0)) i++;

            if (i == p) { // p ist eine Primzahl
                while (x % p == 0) {
                    System.out.print(p + " ");
                    x /= p;    // x = x / p;
                }
            }
            p++;
        }

        System.out.print("\r\n"); // Leerzeile

    }
}
```



```
public static void main(String[] args)
    throws java.io.IOException {

    BufferedReader in;
    String          line;
    int             x;

    System.out.println("Primfaktorenzerlegung");

    in = new BufferedReader(
        new InputStreamReader(System.in));

    line = in.readLine();          // java.io.IOException
    while (line != null) {
        x = Integer.parseInt(line); // NumberFormatException
        PrintPrimFaktoren(x);
        line = in.readLine();
    }
}
}
```

Eingabe:	Ausgabe:
	Primfaktorenzerlegung
1	Primfaktoren von 1:
2	Primfaktoren von 2: 2
3	Primfaktoren von 3: 3
4	Primfaktoren von 4: 2 2
5	Primfaktoren von 5: 5
6	Primfaktoren von 6: 2 3
7	Primfaktoren von 7: 7
8	Primfaktoren von 8: 2 2 2
9	Primfaktoren von 9: 3 3
10	Primfaktoren von 10: 2 5
11	Primfaktoren von 11: 11
12	Primfaktoren von 12: 2 2 3
13	Primfaktoren von 13: 13
14	Primfaktoren von 14: 2 7
15	Primfaktoren von 15: 3 5
16	Primfaktoren von 16: 2 2 2 2
17	Primfaktoren von 17: 17
18	Primfaktoren von 18: 2 3 3
19	Primfaktoren von 19: 19
20	Primfaktoren von 20: 2 2 5
21	Primfaktoren von 21: 3 7
25	Primfaktoren von 25: 5 5
125	Primfaktoren von 125: 5 5 5
625	Primfaktoren von 625: 5 5 5 5
3125	Primfaktoren von 3125: 5 5 5 5 5
1024	Primfaktoren von 1024: 2 2 2 2 2 2 2 2 2 2
4096	Primfaktoren von 4096: 2 2 2 2 2 2 2 2 2 2 2 2
36	Primfaktoren von 36: 2 2 3 3
99	Primfaktoren von 99: 3 3 11
98	Primfaktoren von 98: 2 7 7
97	Primfaktoren von 97: 97
100	Primfaktoren von 100: 2 2 5 5
102	Primfaktoren von 102: 2 3 17
156	Primfaktoren von 156: 2 2 3 13
658	Primfaktoren von 658: 2 7 47
149	Primfaktoren von 149: 149
243	Primfaktoren von 243: 3 3 3 3 3
236	Primfaktoren von 236: 2 2 59

```
/**
 *
 * Aufgabe:
 * Schreiben Sie ein Programm, das von stdin zeilenweise
 * einliest und die Anzahl der Zeilen berechnet. Das
 * Programm soll am Ende die Anzahl der Zeilen ausgeben.
 *
 * @author Peter René Dietmüller
 * @version 1.0, 1998/09/28
 */

import java.io.*;

public class LineCount {

    public static void main(String[] args)
        throws java.io.IOException {

        BufferedReader in;
        String line;
        int lineCount;

        lineCount = 0;
        in = new BufferedReader(
            new InputStreamReader(System.in));

        line = in.readLine();

        while (line != null) {

            lineCount++;
            line = in.readLine();

        }

        System.out.println("Es wurden " + lineCount +
            " Zeilen gezählt.");

    }

}
```

```
/**
 *
 * Aufgabe:
 * Schreiben Sie ein Programm, das zeilenweise von stdin
 * einliest und die Zeilen mit einer Zeilennummer
 * versehen wieder ausgibt.
 *
 * @author Peter René Dietmüller
 * @version 1.0, 09/98
 */

import java.io.*;

class LineNumber {

    public static void main(String[] args)
        throws java.io.IOException {

        BufferedReader in;
        String         line;
        int            lineCount;

        lineCount = 0;
        in = new BufferedReader(
            new InputStreamReader(System.in));

        line = in.readLine();
        while (line != null) {

            lineCount++;
            System.out.println(lineCount + ":" + line);

            line = in.readLine();

        }

    }

}
```

```

/**
Zeigt das Problem beim shiften des Datentyps byte
@author Dipl.-Ing. Michael Sonntag
@version 1.0
*/
public class ShiftTest
{

public static void main(String[] args)
{
    byte byteVar;
    byteVar=31;
    System.out.println("Byte="+byteVar+
        "\nByte shift signed: "+(byteVar>>2)+
        "\nByte shift unsigned: "+(byteVar>>>2)+"\n\n");
    byteVar=-43;
    System.out.println("Byte="+byteVar+
        "\nByte shift signed: "+(byteVar>>2)+
        "\nByte shift unsigned: "+(byteVar>>>2)+"\n\n");
    System.out.println("Byte="+byteVar+
        "\nByte shift signed & 0x3f: "+
            ((byteVar>>2)&0x3f)+
        "\nByte shift unsigned & 0x3f: "+
            ((byteVar>>>2)&0x3f)+"\n\n");
}
}
}

```

## Ausgabe:

```

Byte=31
Byte shift signed: 7
Byte shift unsigned: 7

```

```

Byte=-43 // = 11010101
Byte shift signed: -11 // = 11110101
Byte shift unsigned: 1073741813
// = 00111111 11111111 11111111 11110101
// Wird automatisch VOR dem Schieben auf int erweitert!

```

```

Byte=-43 // = 11010101
Byte shift signed & 0x3f: 53 // = 00110101
Byte shift unsigned & 0x3f: 53 // = 00110101

```