

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x\%4 \neq 0 \mid \neg(x\%100 == 0 \& \&x\%400 == 0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x\%4 \neq 0 \mid \neg(x\%100 == 0 \& \& x\%400 == 0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)} \quad \text{dabei ist } S \text{ der halbe Umfang}$$
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0 \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger



## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.
- Deklariere Sie das Array mit Namen alpha.
  - Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- c) Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0) \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0)$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger



## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x\%4 \neq 0) \mid \neg(x\%100 == 0 \& \& x\%400 == 0)$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x\%4 \neq 0) \mid \neg(x\%100 == 0 \& \& x\%400 == 0)$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0 \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0 \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger



## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.
- Deklariere Sie das Array mit Namen alpha.
  - Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- c) Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)} \quad \text{dabei ist } S \text{ der halbe Umfang}$$
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \quad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0) \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0)$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
 z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0) \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0)$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0) \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0)$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger



## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \quad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.
- Deklariieren Sie das Array mit Namen alpha.
  - Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- c) Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0) \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0)$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0 \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger



## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)} \quad \text{dabei ist } S \text{ der halbe Umfang}$$
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0) \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0)$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariieren Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x\%4 \neq 0 \mid \neg(x\%100 == 0 \& \& x\%400 == 0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger



## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)} \quad \text{dabei ist } S \text{ der halbe Umfang}$$
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$\neg(x \% 4 \neq 0 \mid \neg(x \% 100 == 0 \ \& \ x \% 400 == 0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger

## Programmieren in Java

1. Welche Ausgabe erzeugt folgendes Programmfragment:

```
byte byteOne;
short shortOne;

shortOne = 383;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 511;
byteOne = (byte) shortOne;
System.out.println(byteOne);

shortOne = 510;
byteOne = (byte) shortOne;
System.out.println(byteOne);
```

Beweisen Sie die Richtigkeit der Aussage durch Rechnung im Dualsystem.

2. Welche Ausgabe erzeugt folgendes Programmfragment:

```
public static void main(String[] args) {
    int erg = funktion1(221, 68);
    System.out.println(erg);
}

private static int funktion1(int z1, int z2) {
    while (z2 != 0) {
        if (z1 > z2) {
            z1 = z1 - z2;
        } else {
            z2 = z2 - z1;
        }
    }
    return z1;
}
```

Erstellen Sie eine Tabelle, die die Werte von z1 und z2 während der Schleifendurchläufe aufzeigt.

3. In einem Zweidimensionalen Array aus Buchstaben soll das Alphabet in Großbuchstaben in 5 Zeichen je Zeile gespeichert werden.

- Deklariere Sie das Array mit Namen alpha.
- Füllen Sie es mit Hilfe von Schleifen. Ungenutzte Elemente des Arrays sollen mit Leerzeichen gefüllt werden.

$$\alpha = \begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \\ Z & & & & \end{bmatrix} \qquad \alpha' = \begin{bmatrix} E & D & C & B & A \\ J & I & H & G & F \\ O & N & M & L & K \\ T & S & R & Q & P \\ Y & X & W & V & U \\ & & & & Z \end{bmatrix}$$

- Zeichnen Sie ein Struktogramm zum Spiegeln der Matrix an der Mittelsenkrechten.

4. Entwerfen Sie in einem UML Klassendiagramm eine Klasse mit Namen Dreieck.
- Legen Sie Attribute fest.
  - Sehen Sie einen geeigneten Konstruktor vor.
  - Sehen Sie eine Methode `isDreieck()` vor, die prüft, ob ein Dreieck vorliegt. Nach der Dreiecksungleichung ist im Dreieck die Summe der Längen zweier Seiten  $a$  und  $b$  stets mindestens so groß wie die Länge der dritten Seite  $c$ . d.h.  $a+b > c$  und  $a+c > b$  und  $b+c > a$ .
  - Sehen Sie eine Methode `isGerade()` vor, die prüft ob alle Eckpunkte auf einer Geraden liegen bzw. ob  $a+b=c$  oder  $a+c=b$  oder  $b+c=a$  ist.
  - Sehen Sie eine Methode `berechneUmfang()` vor.
  - Sehen Sie eine Methode `berechneFläche()` vor.  

$$F = \sqrt{S(S-a)(S-b)(S-c)}$$
*dabei ist  $S$  der halbe Umfang*
  - Schreiben Sie Java Code der Klasse Dreieck.

5. Stellen Sie sich ein 8x8 großes Spielfeld für ein Brettspiel vor. Legen Sie eine Datenstruktur zum Speichern der Feldinhalte an. Initialisieren Die Felder abwechselnd mit 1 und 0, so dass bei der Ausgabe folgendes Muster entsteht:

```

0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0

```

6. Schreiben Sie eine Methode, die ein quadratisches zweidimensionales Array von Ganzzahlen beliebiger Größe erzeugt. Alle Elemente am Rand sollen mit einer 1 gefüllt werden. Die Elemente auf den Diagonalen erhalten den Wert -1, alle anderen Elemente sollen 0 sein.  
z.B. Größe 10:

```

1  1  1  1  1  1  1  1  1  1
1 -1  0  0  0  0  0  0 -1  1
1  0 -1  0  0  0  0 -1  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0  0 -1 -1  0  0  0  1
1  0  0 -1  0  0 -1  0  0  1
1  0 -1  0  0  0  0 -1  0  1
1 -1  0  0  0  0  0  0 -1  1
1  1  1  1  1  1  1  1  1  1

```

7. Verwenden Sie die Regel von de Morgan zur Vereinfachung des folgenden Ausdrucks:  

$$!(x\%4!=0 \mid \mid (x\%100==0 \& \& x\%400==0))$$

8. Will man aus einer .java-Datei eine .class-Datei erzeugen, so braucht man:
- assembler
  - virtual machine
  - compiler
  - debugger