

# Übung 1

## Praktische Informatik I

Tutor: Blagoy Genov

11.11.2011

WiSe 2011/12

### Aufgabe 1 \*<sup>1</sup>

Gegeben ist die Ganzzahl  $a = 32$  vom Typ `short` mit Binärdarstellung:

Bit-Nr.:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bitwert:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Wie kann man  $a$  durch (mehrfache) Anwendung der Operatoren `<<` und `|` auf 3242 erhöhen. Schreibt den entsprechenden **Java-Quellcode**. Die Binärdarstellung von 3242 lautet:

Bit-Nr.:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bitwert:	0	0	0	0	1	1	0	0	1	0	1	0	1	0	1	0

Wie würde die Lösung aussehen, wenn anstatt `<<` der Operator `>>>` zur Verfügung stünde?

Für die Lösung der Aufgabe benötigt ihr die Zahlen 1 und 16384 mit Binärdarstellungen:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16384	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

*Lösung mit `|` und `<<`:*

```
a = (short)( a | ( 1 << 1 ) );  
a = (short)( a | ( 1 << 3 ) );  
a = (short)( a | ( 1 << 7 ) );  
a = (short)( a | ( 1 << 10 ) );  
a = (short)( a | ( 1 << 11 ) );
```

*Lösung mit `|` und `>>>`:*

```
a = (short)( a | ( 16384 >>> 3 ) );  
a = (short)( a | ( 16384 >>> 4 ) );  
a = (short)( a | ( 16384 >>> 7 ) );  
a = (short)( a | ( 16384 >>> 11 ) );  
a = (short)( a | ( 16384 >>> 13 ) );
```

<sup>1</sup>\* Muss jeder können, \*\* Für eine gute Note, \*\*\* Für die ganz harten

## Aufgabe 2 \*

Gegeben ist die Ganzzahl  $a = 15439$  vom Typ `short` mit Binärdarstellung:

Bit-Nr.:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bitwert:	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1

Wie kann man nun  $a$  durch Anwendung der Operatoren `<<`, `&` und `~` zu 3082 reduzieren. Die Binärdarstellung von 3082 lautet:

Bit-Nr.:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bitwert:	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0

Wie würde die Lösung aussehen, wenn anstatt `<<` der Operator `>>>` zur Verfügung stünde?

*Lösung mit `&`, `~` und `<<`:*

```
a = (short)( a & ~( 1 << 0 ) );
a = (short)( a & ~( 1 << 2 ) );
a = (short)( a & ~( 1 << 6 ) );
a = (short)( a & ~( 1 << 12 ) );
a = (short)( a & ~( 1 << 13 ) );
```

*Lösung mit `&`, `~` und `>>>`:*

```
a = (short)( a & ~( 16384 >>> 1 ) );
a = (short)( a & ~( 16384 >>> 2 ) );
a = (short)( a & ~( 16384 >>> 8 ) );
a = (short)( a & ~( 16384 >>> 12 ) );
a = (short)( a & ~( 16384 >>> 14 ) );
```

## Aufgabe 3 \*\*

Der Operator XOR (Exclusive OR) kann folgendermaßen auf Mengen definiert werden:

$$A \text{ xor } B := \{x \mid (x \in A \wedge x \notin B) \vee (x \notin A \wedge x \in B)\}$$

In Java gibt es für diese Operation eine direkte Umsetzung: `^`.

Wie kann man diesen Operator mithilfe der Operatoren `|` (Vereinigung), `&` (Durchschnitt) und `~` (Komplement) darstellen? Reichen nur Vereinigung und Komplement oder nur Durchschnitt und Komplement aus? Wie kann man den Durchschnitt (`&`) durch die Operatoren `|` und `^` ersetzen? Und die Vereinigung (`|`) durch `&` und `^`?

*Lösung:*

```
a ^ b ≡ ( a & ~b ) | ( ~a & b )
a ^ b ≡ ~( ~a | b ) | ~( a | ~b )
a ^ b ≡ ~( ~( a & ~b ) & ~( ~a & b ))
a & b ≡ ( a ^ b ) ^ ( a | b )
a | b ≡ ( a ^ b ) ^ ( a & b )
```

## Aufgabe 4 \*\*

Gegeben ist eine beliebige Ganzzahl  $a$  vom Typ `int`. Wie kann man alle Bits von  $a$  ohne den Operator  $\sim$  invertieren? Setzt den Algorithmus in Java mithilfe einer `for`-Schleife um.

*Lösung 1:*

```
for ( int position = 0; position < 32; position++ ) {  
    a = a ^ ( 1 << position );  
}
```

*Lösung 2:*

```
for ( int position = 0; position < 32; position++ ) {  
    if ( ( a & ( 1 << position ) ) == ( 1 << position ) ) {  
        a = a & ~( 1 << position );  
    } else {  
        a = a | ( 1 << position );  
    }  
}
```

## Aufgabe 5 \*\*\*

Gegeben sind die Ganzzahlen  $a, b, c, d, e$  und  $x$  vom Typ `short` mit folgenden Binärdarstellungen:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$a$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$b$	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
$c$	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1
$d$	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$e$	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
$x$	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1

Wie kann man  $x$  durch  $a, b, c, d, e$  und Exclusive OR ( $\wedge$ ) darstellen? Stellt die entsprechende Formel auf.

*Lösung:*  $x = (\text{short})( a \wedge c \wedge d \wedge e )$