

Laboration 9

Avsikten med laborationen är att få en känsla för hur information kodas i datorns minne. Samtidigt får du träning på att skriva och dokumentera metoder.

Skapa paketet **laboration9** innan du fortsätter med laborationen.

Grundläggande uppgifter

Uppgift 9a

Vilket decimalt värde representerar de binära talen?

128 64 32 16 8 4 2 1 Värde

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

Du kan kontrollera dina svar med en kursare och när du löst Uppgift 9c.

Uppgift 9b

Skriv talen på binär form:

- A. 3
- B. 7
- C. 127
- D. 128
- E. 32
- F. 33
- G. 113
- H. 197

Lösning får du i Uppgift 9c

Uppgift 9c

Hämta filerna **Uppgift9c.java** och **Binary.java** från kurssidan. Du ska komplettera metoden **toBinary** (i klassen **Binary**) med kod. Metoden ska översätta ett tal i intervallet 0-255 till binär form och skriva ut resultatet. Du kan följa nedanstående halvkod vilken löser uppgiften.

Halvkod

Deklarera variabeln *value* av typen *int*. Initiera värdet till 128.

Deklarera variabeln *argument* av typen *int*. Initiera argumentet till värdet i *decimal*, dvs.

```
int argument = decimal;
```

Deklarera variabeln *res* av typen *String*. Initiera värdet till "".

Medan *value* är större än 0 // Medan är en while-loop

Om *decimal* är större än eller lika med *value*

Lägg till "1" i slutet av *res*

Minska *decimal* med *value*, dvs.

```
decimal = decimal - value;
```

Annars

Lägg till "0" i slutet av *res*

Halvera *value*, dvs

```
value = value / 2;
```

Skriv ut *argument = res*

Testa din lösning med **Uppgift9c**. Ett lyckat körresultat ska se ut så här:

```
3 = 00000011
7 = 00000111
127 = 01111111
128 = 10000000
32 = 00100000
33 = 00100001
113 = 01110001
197 = 11000101
```

Uppgift 9d

Komplettera metoden **toDecimal** i klassen **Binary** med kod. Metoden ska översätta ett tal på binär form till ett tal på decimal form (dvs ett heltal) och skriva ut resultatet:

binärt tal = heltal t.ex. 01111111 = 127

Du kan följa nedanstående halvkod för att lösa uppgiften:

Halvkod

Deklarera variabeln *res* av typen *int*. Initiera till värdet 0.

Deklarera variabeln *value* av typen *int*. Initiera till värdet 1.

Itererera från (antal tecken i *binary* - 1) till 0, dvs.

```
for( int i = binary.length()-1; i>=0; i--)
```

Om tecknet i position *i* är '1' // tecknet ges med *binary.charAt(i)*

Öka *res* med *value*

Fördubbla *value*

Skriv ut *binary = res*

Testa din lösning med **Uppgift9d**. Ett lyckat körresultat ska se ut så här:

```
00101110 = 46
00000000 = 0
11111111 = 255
1111111111111111 = 65535
```

Fördjupande uppgifter

Uppgift 9e

Om man vill ta reda på koden för en *char* så kan man konvertera *char*-variabeln till en *int*.

Exempel: Nedanstående rader skriver ut koden för tecknet A

```
char tkn = 'A';  
int kod = (int)tkn;  
System.out.println( kod ); // 65 skrivs ut
```

Komplettera metoden **charCode** i klassen **Binary** med kod. Metoden tar emot ett *String*-objekt som parameter och ska skriva ut tecknen i strängen + deras kod i Output-fönstret.

Testkod

```
Binary prog = new Binary();  
prog.charCode( "solbad" );
```

Körresultat

```
s = 115  
o = 111  
l = 108  
b = 98  
a = 97  
d = 100
```

Uppgift 9f

Ändra metoden **toBinary** så att metoden returnerar ett *String*-objekt i stället för att göra en utskrift i Output-fönstret. Om användaren anropar metoden med ett negativt tal eller ett tal större än 255 ska metoden returnera "-----"

Deklaration: `public String toBinary(int decimal)`

Testkod

```
Binary bin = new Binary();  
String res = bin.toBinary( 9 );  
System.out.println( res );  
System.out.println( bin.toBinary( -3 ) );  
System.out.println( bin.toBinary( 600 ) );
```

Körresultat

```
00001001  
-----  
-----
```

Uppgift 9g

Ändra metoden **toDecimal** så att metoden returnerar en *int* i stället för att göra en utskrift i Output-fönstret.

Testkod

```
Binary bin = new Binary();  
int res = bin.toDecimal( "00001001" );  
System.out.println( res );
```

Körresultat

9

Extrauppgift

Uppgift 9h

Skriv metoden **shortToBinary(int decimal)** i klassen **Binary**. Metoden ska returnera en sträng i binär form. Argumentet vid anropet ska vara i intervallet 0 – 65535.

Försök använda dig av två anrop till toBinary i din lösning.

Testkod

```
Binary bin = new Binary();  
System.out.println( 76 + " = " + bin.shortToBinary(76) );  
System.out.println( 2556 + " = " + bin.shortToBinary(2556) );  
System.out.println( 64111 + " = " + bin.shortToBinary(64111) );  
System.out.println( u3.toDecimal( bin.shortToBinary(76) ) );  
System.out.println( u3.toDecimal( bin.shortToBinary(2556) ) );  
System.out.println( u3.toDecimal( bin.shortToBinary(64111) ) );
```

Körresultat

```
76 = 0000000001001100  
2556 = 0000100111111100  
64111 = 1111101001101111  
76  
2556  
64111
```

Lösningsförslag

```
public void toBinary( int decimal ) {
    int value = 128, argument = decimal;
    String res = "";
    while (value > 0) {
        if (decimal >= value ) {
            res += "1";
            decimal -= value;
        } else {
            res += "0";
        }
        value /= 2;
    }
    System.out.println(argument + " = " + res);
}
-----

public void toDecimal( String binary ) {
    int res = 0, value = 1;
    for( int i = binary.length()-1; i>=0; i-- ) {
        if( binary.charAt( i ) == '1' )
            res += value;
        value *= 2;
    }
    System.out.println( binary + " = " + res );
}
-----

public void charCode( String str ) {
    for( int i = 0; i < str.length(); i++ ) {
        System.out.println( str.charAt( i ) + " = " + (int)str.charAt( i ) );
    }
}
-----

// Ny version
public String toBinary( int decimal ) {
    int value = 128;
    String res = "";
    if( (decimal >= 0) && (decimal <= 255) ) {
        while (value > 0) {
            if (decimal >= value ) {
                res += "1";
                decimal -= value;
            } else {
                res += "0";
            }
            value /= 2;
        }
    } else {
        res = "-----";
    }
    return res;
}
-----
```

```
// Ny version
public int toDecimal( String binary ) {
    int res = 0, value = 1;
    for( int i = binary.length()-1; i>=0; i-- ) {
        if( binary.charAt( i ) == '1' )
            res += value;
        value *= 2;
    }
    return res;
}
-----

public String shortToBinary(int decimal) {
    return byteToBinary( decimal / 256 ) + byteToBinary( decimal % 256 );
}
```