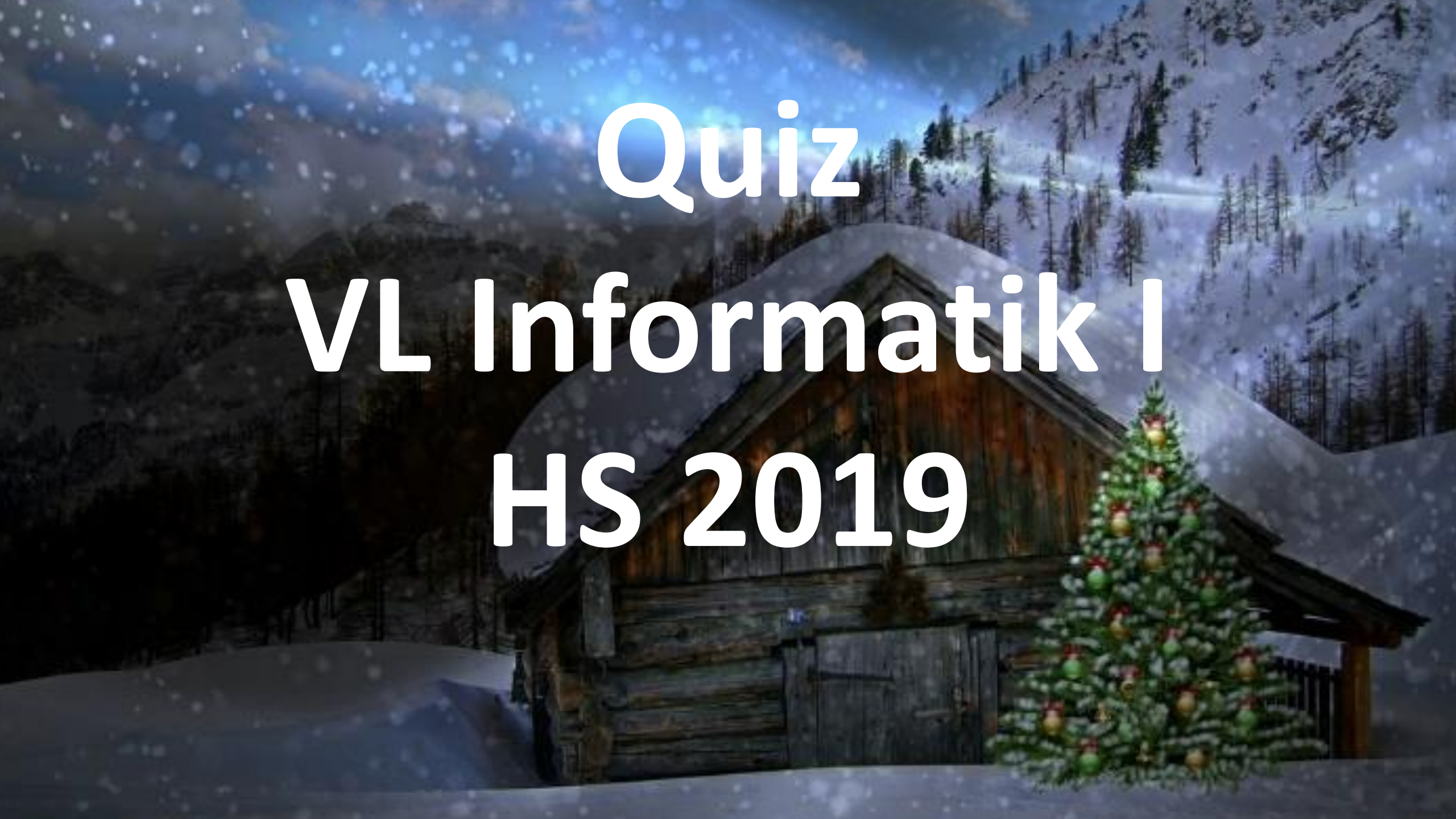
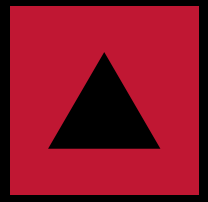

Quiz
VL Informatik I
HS 2019

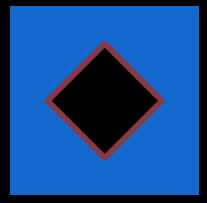
Vorlesung 2: Zimtsterne

Ganze Zahlen, Arithmetische Ausdrücke

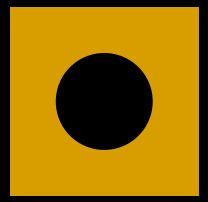
```
// 11 Zimsterne
int zimtsterne = 11;
Out.println("Zutaten : " +
  zimtsterne / 10 + " Eiweiss, "
  + 1 + zimtsterne * 0.05 + " EL Zimt." );
```



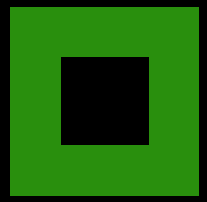
Zutaten: 1 Eiweiss, 1.55 TL Zimt.



Zutaten: 1.1 Eiweiss, 1.55 TL Zimt.



Zutaten: 1 Eiweiss, 10.55 TL Zimt.



Zutaten: 1.1 Eiweiss, 10.55 TL Zimt.

```
// 11 Zimsterne
int zimtsterne = 11;
Out.println("Zutaten : " +
  zimtsterne / 10 + " Eiweiss, "
  + 1 + zimtsterne * 0.05 + " EL Zimt." );
```

Zutaten: 1 Eiweiss,
10.55 TL Zimt.

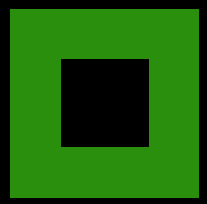
Vorlesung 3: Wir können es nicht erwarten

Typen float und double

```java
// Ungeduld ?
double prozent = 100;
for (int tag = 24; tag > 0; --tag) {
  prozent -= 100.0 / 24;
}
if (prozent == 0){
  Out.println("Frohe Weihnachten!");
} else{
  Out.println("Leider keine Geschenke.");
}
```

- "Frohe Weihnachten"
- Terminiert nicht.
- Compiler Fehlermeldung
- "Leider keine Geschenke"

```java
// Ungeduld ?
double prozent = 100;
for (int tag = 24; tag > 0; --tag) {
  prozent -= 100.0 / 24;
}
if (prozent == 0){
  Out.println("Frohe Weihnachten!");
} else{
  Out.println("Leider keine Geschenke.");
}
```



"Leider keine Geschenke"

Vorlesung 4: Die Weihnachtsformel

Typen `float` und `double`

$$\text{Number of ornaments} = \frac{\sqrt{17}}{20} \times (\text{Tree height in cms})$$

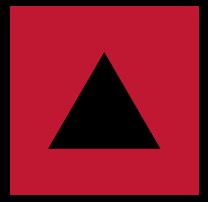
$$\text{Length of tinsel (cms)} = \frac{13 \times \pi}{8} \times (\text{Tree height in cms})$$

$$\text{Length of lights (cms)} = \pi \times (\text{Tree height in cms})$$

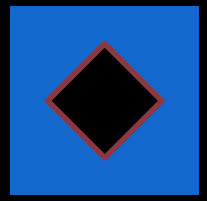
$$\text{Height of the star/angel (cms)} = \frac{(\text{Tree height in cms})}{10}$$

```
6
10.5
13.875
16.4062
18.3047
19.7285
20.7964
21.5973
22.198
22.6485
22.9864
23.2398
23.4298
23.5724
23.6793
23.7595
23.8196
23.8647
23.8985
23.9239
23.9429
23.9572
23.9679
23.9759
23.9819
23.9865
23.9898
23.9924
23.9943
23.9957
23.9968
23.9976
23.9982
23.9986
23.999
23.9992
23.9994
23.9996
23.9997
23.9998
23.9998
23.9999
23.9999
23.9999
23.9999
24
```

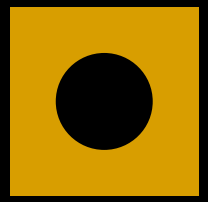
```
// Formel für die Weihnachtszahl
double nikolaus_tag = 6.0;
double heilige_koenige = 3.0;
double adventssonntage = 4.0;
double x = nikolaus_tag;
double w = 0.0;

while (x > 1e-5) {
   w += x;
   x *= heilige_koenige / adventssonntage;
}

Out.println("W = " + (int)(w + 0.5));
```



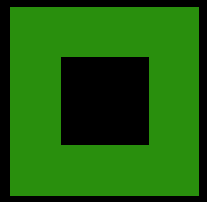
W = 7



W = 24

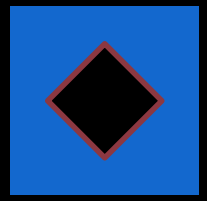


W = 12



W = 42

```
// Formel für die Weihnachtszahl
double nikolaus_tag = 6.0;
double heilige_koenige = 3.0;
double adventssonntage = 4.0;
double x = nikolaus_tag;
double w = 0.0;

while (x > 1e-5) {
   w += x;
   x *= heilige_koenige / adventssonntage;
}

Out.println("W = " + (int)(w + 0.5));
```



W = 24

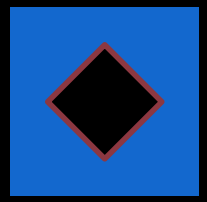
```java
// Formel für die Weihnachtszahl
double nikolaus_tag = 6.0;
double heilige_koenige = 3.0;
double adventssonntage = 4.0;
double x = nikolaus_tag;
double w = 0.0;

while (x > 1e-5) {
   w += x;
   x *= heilige_koenige / adventssonntage;
}

Out.println("W = " + (int)(w + 0.5));
```

Lösung:

$6 \cdot \sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i = 24$

mit geometrischer Reihe

$\sum_{i=0}^{\infty} p^i = \frac{1}{1-p}, 0 \leq p < 1$



```
W = 24
```

Vorlesung 4: Samichlaus-Checker

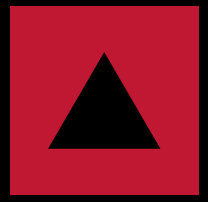
Logische Operatoren, **`if-else`**

```
War Ihr Kind brav? n
Wie alt ist Ihr Kind? 7
Strafe vom Schmutzli!
```

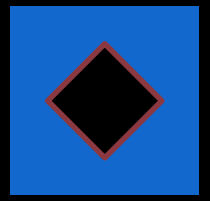
```java
// Samichlaus Checker
Out.print("War Ihr Kind brav (j/n)?");
boolean brav = In.readChar() != 'n';

Out.print("Wie alt ist das Kind?");
int alter = In.readInt();

if                                                    {
   Out.println("Geschenk vom Samichlaus!");
} else {
   Out.println("Strafe vom Schmutzli!");
}
```

*entweder älter als 3 **oder** brav*

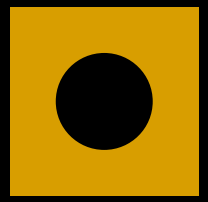
was darf da hin?



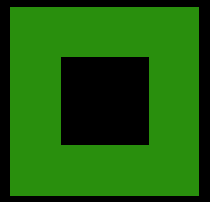
```
brav || alter > 3
```



```
brav != alter > 3
```



```
brav == alter > 3
```

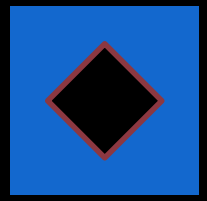


```
brav && alter <= 3
```

```java
// Samichlaus Checker
Out.print("War Ihr Kind brav (j/n)?");
boolean brav = In.readChar() != 'n';

Out.print("Wie alt ist das Kind?");
int alter = In.readInt();

if                                                    {
   Out.println("Geschenk vom Samichlaus!");
} else {
   Out.println("Strafe vom Schmutzli!");
}
```

__entweder__ älter als 3 __oder__ brav

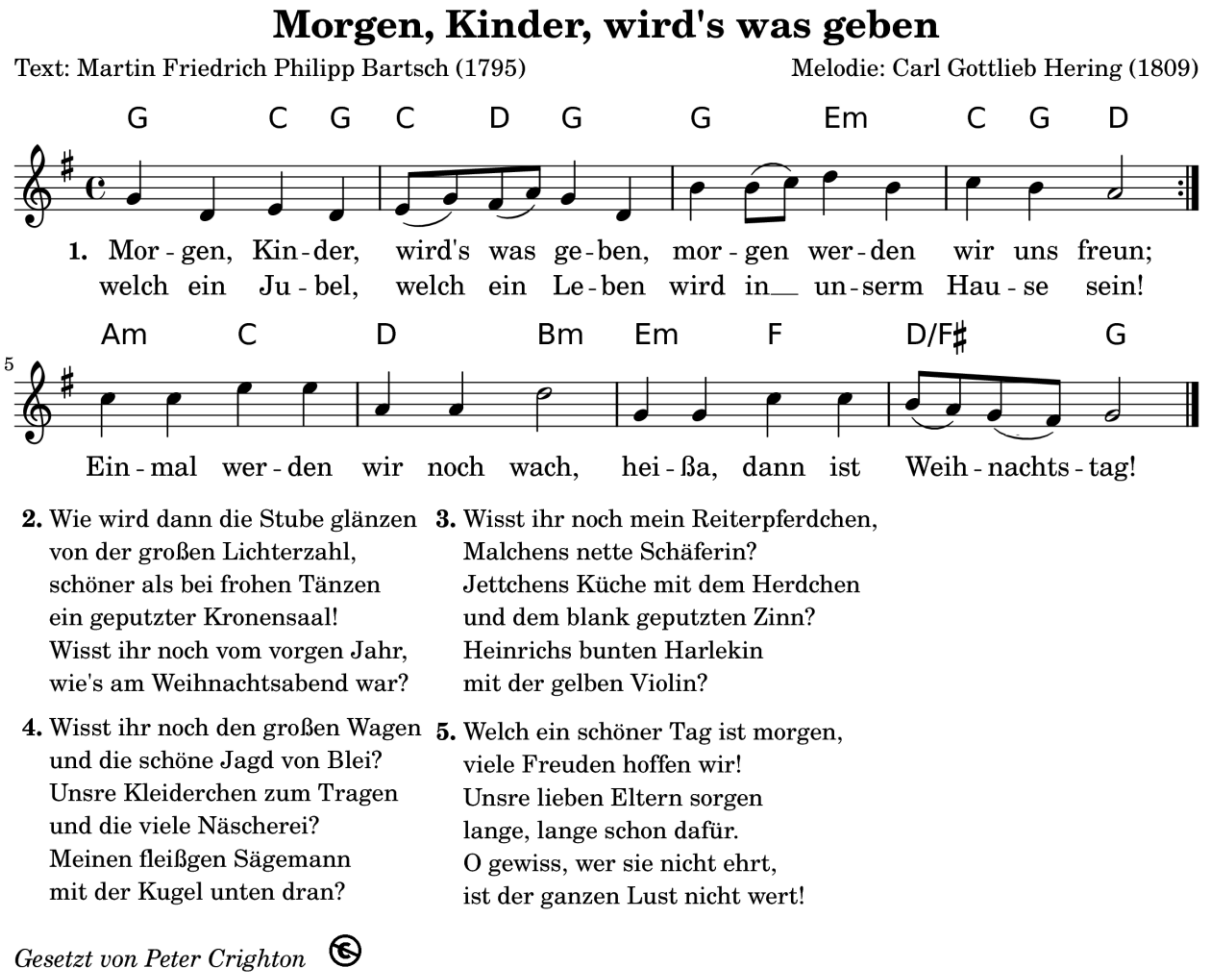
was darf da hin?



```
brav != alter > 3
```

Vorlesung 5: Einmal werden wir noch wach

Arrays und Referenzen



Morgen, Kinder, wird's was geben

Text: Martin Friedrich Philipp Bartsch (1795) — Melodie: Carl Gottlieb Hering (1809)

G C G C D G G Em C G D

1. Mor - gen, Kin - der, wird's was ge - ben, mor - gen wer - den wir uns freun;
welch ein Ju - bel, welch ein Le - ben wird in un - serm Hau - se sein!

Am C D Bm Em F D/F♯ G

Ein - mal wer - den wir noch wach, hei - ßa, dann ist Weih - nachts - tag!

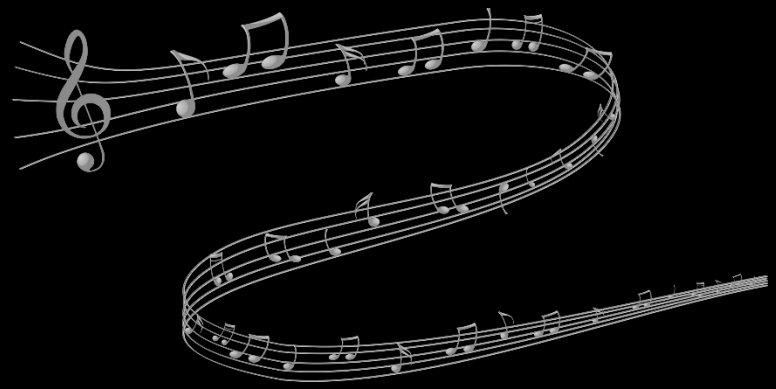
2. Wie wird dann die Stube glänzen
von der großen Lichterzahl,
schöner als bei frohen Tänzen
ein geputzter Kronensaal!
Wisst ihr noch vom vorgen Jahr,
wie's am Weihnachtsabend war?

3. Wisst ihr noch mein Reiterpferdchen,
Malchens nette Schäferin?
Jettchens Küche mit dem Herdchen
und dem blank geputzten Zinn?
Heinrichs bunten Harlekin
mit der gelben Violin?

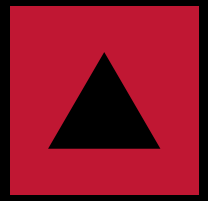
4. Wisst ihr noch den großen Wagen
und die schöne Jagd von Blei?
Unsre Kleiderchen zum Tragen
und die viele Näscherei?
Meinen fleißgen Sägemann
mit der Kugel unten dran?

5. Welch ein schöner Tag ist morgen,
viele Freuden hoffen wir!
Unsre lieben Eltern sorgen
lange, lange schon dafür.
O gewiss, wer sie nicht ehrt,
ist der ganzen Lust nicht wert!

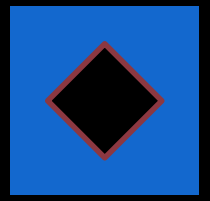
Gesetzt von Peter Crighton



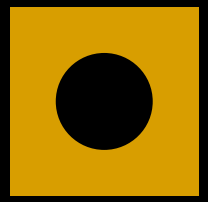
```
// Einmal werden wir noch wach

   public static void swap (String[] lied, int p1, int p2){
      String wort = lied[p1];
      lied[p1] = lied[p2];
      lied[p2] = wort;
   }

   public static void main(String[] args){
      String[] lied = {"Einmal", "werden", "wach", "noch", "wir"};
      swap(lied, 2, 4);
      for(String wort: lied){
         Out.print(wort + " ");
      }
   }
```



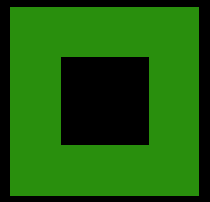
Einmal werden wir noch wach



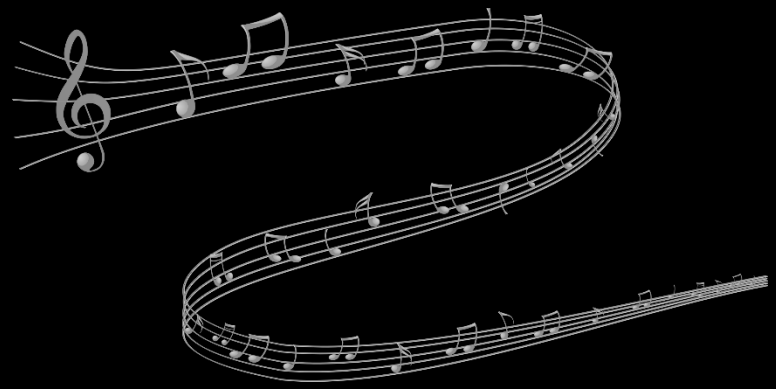
Einmal noch werden wir wach



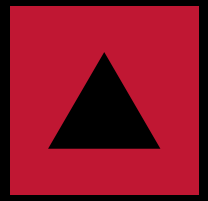
Einmal noch wir werden wach



Einmal wir werden noch wach



```
// Einmal werden wir noch wach

   public static void swap (String[] lied, int p1, int p2){
      String wort = lied[p1];
      lied[p1] = lied[p2];
      lied[p2] = wort;
   }

   public static void main(String[] args){
      String[] lied = {"Einmal", "werden", "wach", "noch", "wir"};
      swap(lied, 2, 4);
      for(String wort: lied){
         Out.print(wort + " ");
      }
   }
```



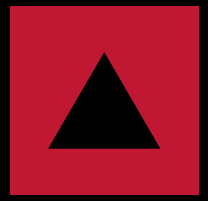
Einmal werden wir noch wach

Vorlesung 4&5: Adventskalender
Arrays und do-Anweisung

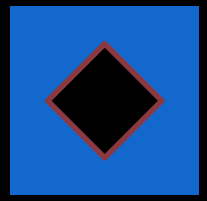

```
Welches Türchen?
1
  Schööön!
Welches Türchen?
2
  Schööön!
Welches Türchen?
4
  Falsches Türchen!
Welches Türchen?
```


```
// Adventskalender
boolean[] offen = new boolean[25];
offen[0] = true;
int tag;
do {
   Out.println("Welches Tuerchen?");
   tag = In.readInt();
   if (          das jeweils nächste Türchen wird geöffnet          ){
      offen[tag] = true;
      Out.println(" Ahhhhh schoen!");
   } else {
      Out.println(" Falsches Tuerchen!");
   }
} while (!offen[24]);
```

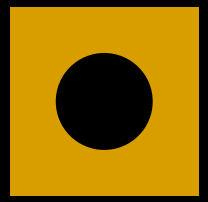
Was muss hier hin?



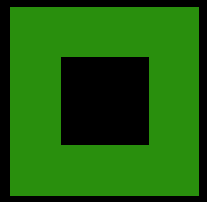
```
!offen[tag] && offen[tag-1]
&& tag > 0 && tag < 25
```



```
tag > 0 && offen[tag-1]
&& tag < 25 && !offen[tag]
```



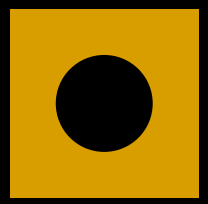
```
tag > 0 && tag < 25
&& !offen[tag] && offen[tag-1]
```



```
egal, alle richtig
```


```
// Adventskalender
boolean[] offen = new boolean[25];
offen[0] = true;
int tag;
do {
   Out.println("Welches Tuerchen?");
   tag = In.readInt();
   if (          das jeweils nächste Türchen wird geöffnet          ){
      offen[tag] = true;
      Out.println(" Ahhhhh schoen!");
   } else {
      Out.println(" Falsches Tuerchen!");
   }
} while (!offen[24]);
```

Was muss hier hin?



```
tag > 0 && tag < 25
&& !offen[tag] && offen[tag-1]
```

Vorlesung 6: Weihnachtsbaum

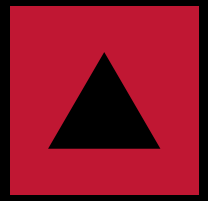
Methoden, Stepwise Refinement

```
*
***
*****
*******
*********
***********
*************
***************
*****************
*******************
```

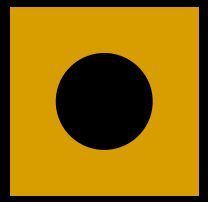
```java
// Weihnachtsbaum
static void draw(char zeichen, int n){
    while(--n >= 0){
        Out.print(zeichen);
    }
}

static void schnitt(int left, int right){
    draw(' ', left);
    draw('*', right);
    Out.println();
}

static void weihnachtsbaum (int hoehe){
    for (int i = 0; i < hoehe; ++i){
        was muss hier hin?
    }
}
```

Ausgabe von weihnachtsbaum(10)

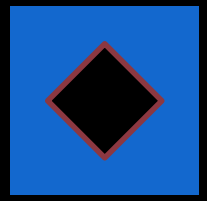
```
         *
        ***
       *****
      *******
     *********
    ***********
   *************
  ***************
 *****************
*******************
```



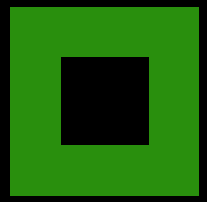
schnitt(i,2*i+1);



schnitt(2*i+1,i);



schnitt(hoehe-i,2*i+1);

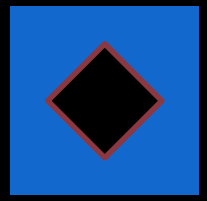


schnitt(hoehe-2*i+1,i);

```java
// Weihnachtsbaum
static void draw(char zeichen, int n){
    while(--n >= 0){
        Out.print(zeichen);
    }
}

static void schnitt(int left, int right){
    draw(' ', left);
    draw('*', right);
    Out.println();
}

static void weihnachtsbaum (int hoehe){
    for (int i = 0; i < hoehe; ++i){
        was muss hier hin?
    }
}
```

Ausgabe von weihnachtsbaum(10)

```
         *
        ***
       *****
      *******
     *********
    ***********
   *************
  ***************
 *****************
*******************
```

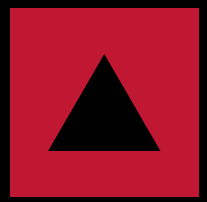


```java
schnitt(hoehe-i,2*i+1);
```

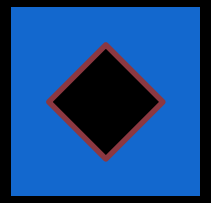
Vorlesung 7: Liedtext Lernen

Rekursion

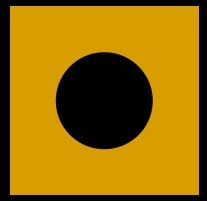
```
public static void Singe(String[] lied, int von, int bis){
  if (von < bis){
    String a = lied[von];
    String b = lied[bis-1];
    Singe(lied,von+1,bis-1);
    Out.print(a + " ");
    Out.print(b + " ");
  }
}

public static void main(String[]args){
  String[] lied = {"O","Froehliche", "Du", "O", "Selige","Du"};
  Singe(lied, 0, 6);
}
```



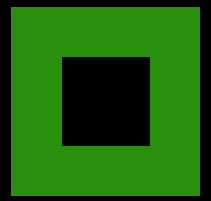
O Du Froehliche
O Du Selige



Du O Froehliche
Selige O Du

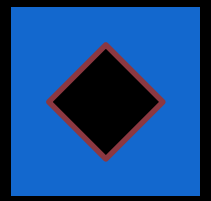


Du Selige O
Du Froehliche O



O Du Froehliche
Du O Selige

```
public static void Singe(String[] lied, int von, int bis){
  if (von < bis){
    String a = lied[von];
    String b = lied[bis-1];
    Singe(lied,von+1,bis-1);
    Out.print(a + " ");
    Out.print(b + " ");
  }
}

public static void main(String[]args){
  String[] lied = {"O","Froehliche", "Du", "O", "Selige","Du"};
  Singe(lied, 0, 6);
}
```



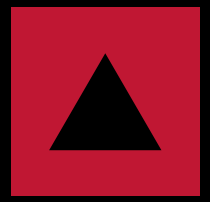
Du O Froehliche
Selige O Du

Vorlesung 8: Rudolph

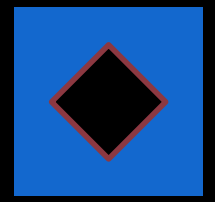
Klassen, Members, Konstruktoren

```java
class Rentier{
   String nasenfarbe;
   public Rentier(){
      nasenfarbe = "braun";
   }
   public Rentier(String farbe){
      nasenfarbe = farbe;
   }
   public String nase(){
      return nasenfarbe;
   }
   public void tausch(Rentier anderes){
      String n = nase();
      nasenfarbe = anderes.nase();
      anderes.nasenfarbe = n;
   }
}
```

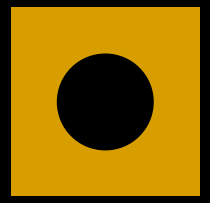
```java
Rentier dasher = new Rentier();
Rentier dancer = new Rentier();
Rentier rudolph = new Rentier("rot");
rudolph.tausch(dancer);
Out.println(dasher.nase());
Out.println(dancer.nase());
Out.println(rudolph.nase());
```



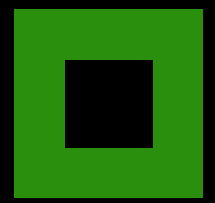
braun
braun
rot



rot
braun
rot



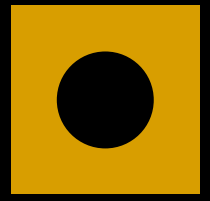
braun
rot
braun



braun
rot
rot

```java
class Rentier{
   String nasenfarbe;
   public Rentier(){
      nasenfarbe = "braun";
   }
   public Rentier(String farbe){
      nasenfarbe = farbe;
   }
   public String nase(){
      return nasenfarbe;
   }
   public void tausch(Rentier anderes){
      String n = nase();
      nasenfarbe = anderes.nase();
      anderes.nasenfarbe = n;
   }
}
```

```java
Rentier dasher = new Rentier();
Rentier dancer = new Rentier();
Rentier rudolph = new Rentier("rot");
rudolph.tausch(dancer);
Out.println(dasher.nase());
Out.println(dancer.nase());
Out.println(rudolph.nase());
```



```
braun
rot
braun
```

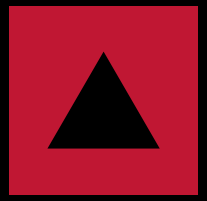
Vorlesungen 9/10: Vererbung und Polymorphie

Der Weihnachtsmann

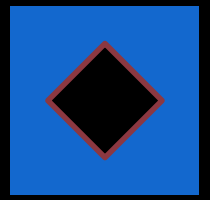
```java
class Weihnachtsmann {
  boolean kuesstMama = false;

  void hoho(){
    Out.println("Hohoho (" + stimme() + ")");
  }

  String stimme(){
    return "beeindruckende Stimme";
  }

  boolean kuesstPapasFrau(){
    return kuesstMama;
  }
}
```

```java
class Papa extends Weihnachtsmann {

  public Papa(){
    kuesstMama = true;
  }

  String stimme(){
    return "verstellte Stimme";
  }
}
```

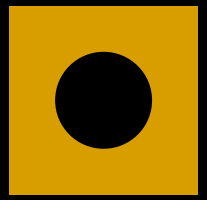

```java
Papa p = new Papa();
Weihnachtsmann w = p;

w.hoho();

if (!w.kuesstPapasFrau()){
  Out.println("Oh lieber Weihnachtsmann!");
} else {
  Out.println("Du bist entlarvt!");
}
```



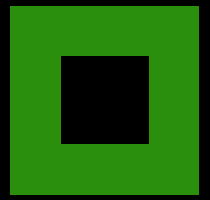
Hohoho
(beeindruckende Stimme)
Oh lieber Weihnachtsmann



Hohoho
(verstellte Stimme)
Oh lieber Weihnachtsmann



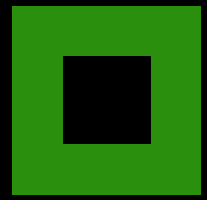
Hohoho
(beeindruckende Stimme)
Du bist entlarvt!



Hohoho
(verstellte Stimme)
Du bist entlarvt!

```java
class Weihnachtsmann {
  boolean kuesstMama = false;

  void hoho(){
    Out.println("Hohoho (" + stimme() + ")");
  }

  String stimme(){
    return "beeindruckende Stimme";
  }

  boolean kuesstPapasFrau(){
    return kuesstMama;
  }
}

class Papa extends Weihnachtsmann {

  public Papa(){
    kuesstMama = true;
  }

  String stimme(){
    return "verstellte Stimme";
  }
}
```


```java
Papa p = new Papa();
Weihnachtsmann w = p;

w.hoho();

if (!w.kuesstPapasFrau()){
  Out.println("Oh lieber Weihnachtsmann!");
} else {
  Out.println("Du bist entlarvt!");
}
```



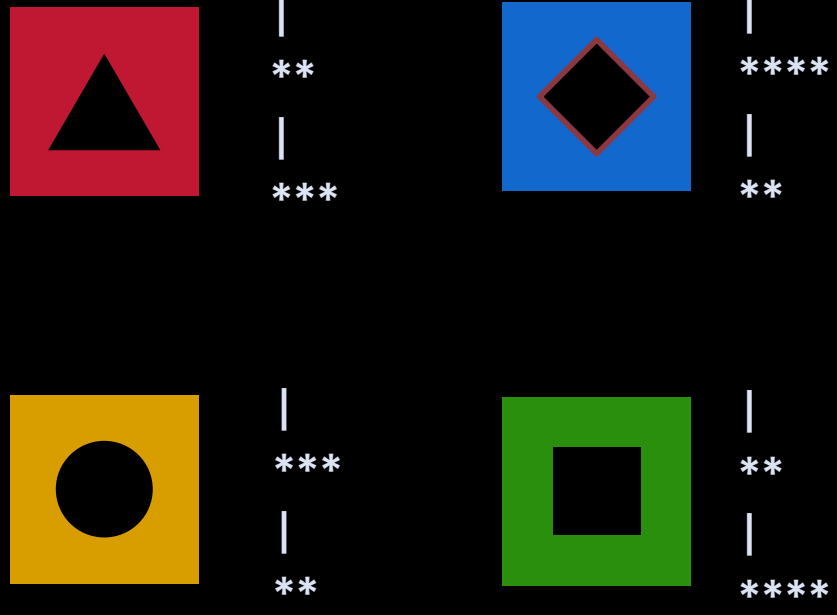
```
Hohoho
(verstellte Stimme)
Du bist entlarvt!
```

Vorlesung 11: Lichterkette

Dynamische Datentypen

```
class Kette{
  String lichter="*";
  Kette kupplung;

  Kette(int len){
    while(--len > 0){
      lichter += lichter;
    }
  }
}

class Vorhang{
  private Kette steckDose = null;

  public void pimpen(Kette neu){
    neu.kupplung = steckDose;
    steckDose = neu;
  }

  public void bestaunen(){
    Kette k = steckDose;
    while (kette != null)
      Out.println("|");
      Out.println(k.lichter);
      kette = kette.kupplung;
    }
  }
}
```

```
public static void main(String[] args){
   Vorhang v = new Vorhang;
   v.pimpen(new Kette(3));
   v.pimpen(new Kette(2));
   v.bestaunen();
}
```



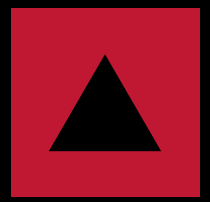
```
class Kette{
  String lichter="*";
  Kette kupplung;

  Kette(int len){
    while(--len > 0){
      lichter += lichter;
    }
  }
}

class Vorhang{
  private Kette steckDose = null;

  public void pimpen(Kette neu){
    neu.kupplung = steckDose;
    steckDose = neu;
  }

  public void bestaunen(){
    Kette k = steckDose;
    while (kette != null)
      Out.println("|");
      Out.println(k.lichter);
      kette = kette.kupplung;
    }
  }
}
```

```
public static void main(String[] args){
   Vorhang v = new Vorhang;
   v.pimpen(new Kette(3));
   v.pimpen(new Kette(2));
   v.bestaunen();
}
```

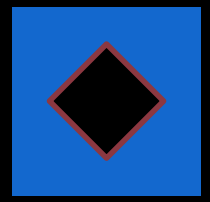
```
|
**
|
****
```

Vorlesung 11: Bescherung
Java Collections

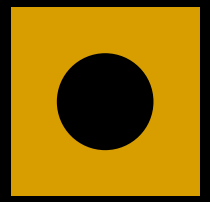
```
Map<String,Integer> w = new TreeMap<String,Integer>();
w.put("Socken",12);
w.put("Holzeisenbahn",35);
w.put("iPhone",800);
w.put("Barbie",25);
for (String a: w.keySet()){
 Out.println(a + ":" + w.get(a));
}
```



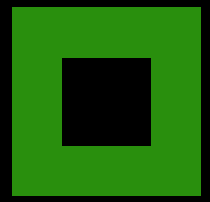
12:Socken
25:Barbie
35:Holzeisenbahn
800:iPhone



Socken:12
Holzeisenbahn:25
iPhone:800
Barbie:35

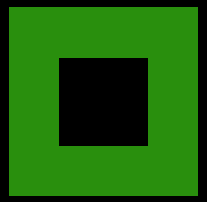


Socken:12
Barbie:25
Holzeisenbahn:35
iPhone:800



Barbie:25
Holzeisenbahn:35
Socken:12
iPhone:800

```
Map<String,Integer> w = new TreeMap<String,Integer>();
w.put("Socken",12);
w.put("Holzeisenbahn",35);
w.put("iPhone",800);
w.put("Barbie",25);
for (String a: w.keySet()){
 Out.println(a + ":" + w.get(a));
}
```



```
Barbie:25
Holzeisenbahn:35
Socken:12
iPhone:800
```