

VL01, Lösung 1

- Informationen in Form von Text, Bildern, Tönen, Programmen oder Zahlen werden durch eine Folge von 0 und 1 codiert. Das Ergebnis dieser Umformung sind die Daten.
- Ein Bit ist die kleinste Informationseinheit und kann die zwei Werte 0 und 1 annehmen. Ein Byte sind 8 Bit.
- Programme und Daten liegen gemeinsam im Hauptspeicher.

VL01, Lösung 2

- a) 9 Stellen, denn $2^8 = 256 < 423 < 512 = 2^9$.
(Bei n Stellen können Zahlen im Bereich 0 bis $2^n - 1$ dargestellt werden.)
- b) Binär: $1101_{10} = 10001001101_2$ Hexadezimal: $0100\ 0100\ 1101_2 = 44D_{16}$

Rechenschema:

```

1101 : 2 = 550 Rest 1
550 : 2 = 275 Rest 0
275 : 2 = 137 Rest 1
137 : 2 =  68 Rest 1
 68 : 2 =  34 Rest 0
 34 : 2 =  17 Rest 0
 17 : 2 =   8 Rest 1
  8 : 2 =   4 Rest 0
  4 : 2 =   2 Rest 0
  2 : 2 =   1 Rest 0
  1 : 2 =   0 Rest 1

```

$$\begin{array}{l} 1101 : 16 = 68 \text{ Rest } \mathbf{13 (= D)} \\ 68 : 16 = 4 \text{ Rest } \mathbf{4} \\ 4 : 16 = 0 \text{ Rest } \mathbf{4} \end{array}$$

VL01, Lösung 3

Rechenschema für 0,1:

$$\begin{array}{l} 0,1 \cdot 2 = 0,2 \\ 0,2 \cdot 2 = 0,4 \\ 0,4 \cdot 2 = 0,8 \\ 0,8 \cdot 2 = 1,6 \\ 0,6 \cdot 2 = 1,2 \\ 0,2 \cdot 2 = 0,4 \end{array}$$

Die Binärdarstellung ist periodisch und lautet $0,000\overline{11}$.

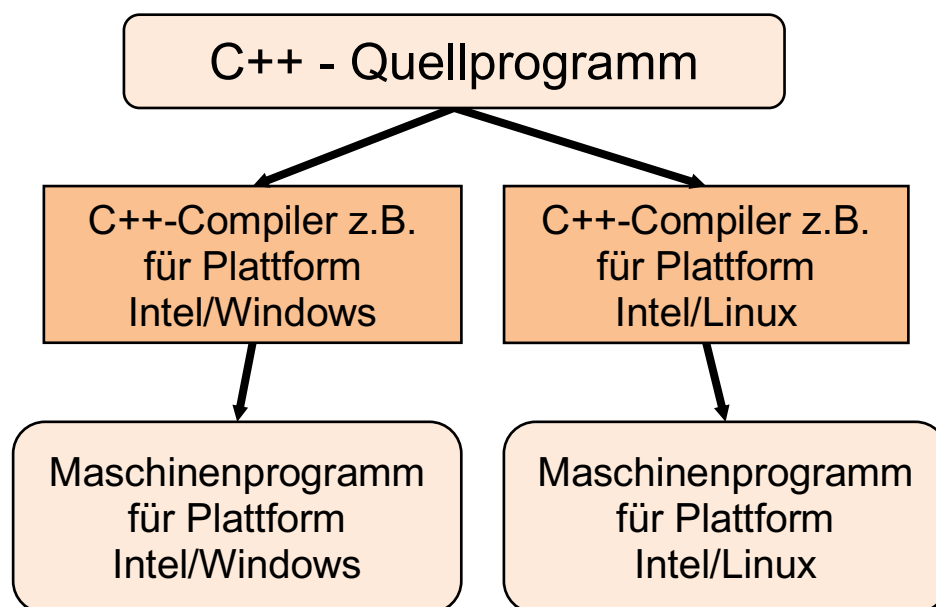
Damit ist auch klar, dass $0,2$, $0,4$, $0,6$ und $0,8$ keine endliche Darstellung besitzen können, da sie Teil der Periode sind. Für $0,3$ ergibt sich nach der ersten Multiplikation mit 2 die Zahl $0,6$. Für $0,7$ ergibt sich $1,4$ und anschließend $0,4$. Für $0,9$ ergibt sich $1,8$ und anschließend $0,8$. Diese können somit auch keine endliche Binärdarstellung besitzen. Nur $0,5_{10}$ besitzt die endliche Binärdarstellung $0,1_2$.

VL01, Lösung 4

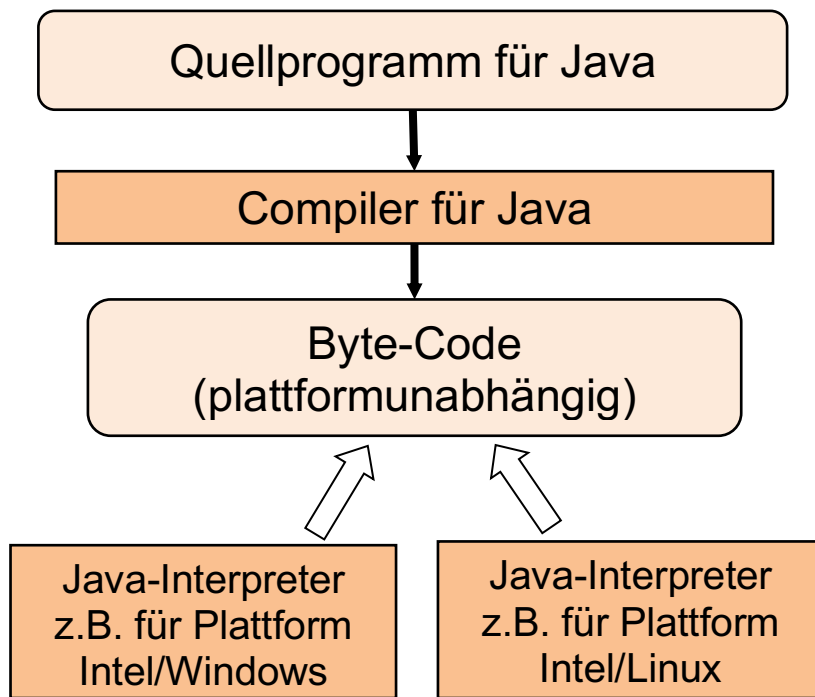
- a) Der Compiler dient zur Übersetzung von Programmen einer Quellsprache in eine Zielsprache, wobei hingegen der Interpreter ein Programm einer anderen Programmiersprache analysiert und unmittelbar ausführt.
- b) Der Just-in-time-Compiler dient zur beschleunigten Ausführung des Byte-Codes, da dieser Byte-Code-Sequenzen bereits während der Laufzeit in Maschinencode übersetzt. Dabei ist jedoch zu beachten, dass auch dieser Compiler plattformabhängig ist und für jede Plattform ein eigener Just-in-time-Compiler benötigt wird.

VL01, Lösung 5

C++-Quellcode benötigt einen Compiler z.B. für die Plattform Intel/Windows und einen zweiten Compiler für die Plattform Intel/Linux. Erzeugt werden also zwei Maschinenprogramme für die jeweilige Plattform.



Das Quellprogramm in Java muss nur mit einem Compiler übersetzt werden und man erhält plattformunabhängigen Byte-Code. Für jede Plattform benötigt man nun jeweils einen Interpreter, der den Byte-Code auf der jeweiligen Plattform ausführt.



VL01, Lösung 6

```
class Hello
{
    public static void main(String[] args)
    {
        System.out.println("Max Mustermann 7090000");
    }
}
```