

2.3 Übersetzungsarten

Donnerstag, 1. Dezember 2022 08:14

2.3. Übersetzungsarten

Neben den oben erwähnten höheren Programmiersprachen gibt es auch niedrigere Programmiersprachen. Vereinfacht ausgedrückt, können Computer nur Programme ausführen, die in niederen Programmiersprachen geschrieben wurden. Programme, die in höheren Programmiersprachen geschrieben wurden, müssen deshalb zunächst verarbeitet werden, bevor sie ausgeführt werden können. Diese Verarbeitung benötigt ein bisschen Zeit, was ein kleiner Nachteil von höheren Programmiersprachen ist.

Natürlich haben höhere Programmiersprachen aber auch einige Vorteile:

- sie lassen sich schneller schreiben
- sie sind kürzer und einfacher zu lesen
- sie sind mit größerer Wahrscheinlichkeit richtig
- sie sind portierbar, d.h. Programme, die mit ihnen geschrieben wurden, können auf nur wenigen oder gar keinen Änderungen auf verschiedenen Arten von Computern ausgeführt werden

(mit niederen Programmiersprachen geschriebene Programme können nur auf einer Art von Computern ausgeführt werden und müssen für andere Computertypen neu geschrieben werden.)

Für die Verarbeitung von höheren in niedrigere Programmiersprachen sind zwei Arten von Programmen erforderlich: **Interpreter und Compiler**



10 min.



1. Beschreiben Sie den Unterschied zwischen einem Compiler und Interpreter
2. Stellen Sie die Funktionsweise von Interpreter und Compiler grafisch dar.



Du möchtest dir den Unterschied zwischen einem Compiler und Interpreter nochmal in Ruhe ansehen? Dann scanne den QR-Code ab, dort findest du ein Video.

➤ **Unterschied: Compiler und Interpreter**

1. Funktionen von Interpreter und Compiler.

Interpreter	Compiler
	
<ul style="list-style-type: none"> •übersetzt den Quellcode zeilenweise in Maschinencode •in Jeder Zeile erfolgt eine Syntaxprüfung •Der Interpreter ist daher langsamer als ein Compiler 	<ul style="list-style-type: none"> •übersetzt den Code im Ganzen in Maschinsprache •arbeitet i.d.R. schneller als der Interpreter •Fehler werden erst am Ende ausgegeben

2. Stellen Sie die Funktionsweise von Interpreter und Compiler grafisch dar.



Interpreter

Code → Interpreter → Output



Compiler

Code → Compiler → Maschinencode → Output