# Compilieren zur direkten oder indirekten Ausführung auf einer Maschine

PC mit Windows

Mac

Unix:   
Linux, Ubuntu, Solaris

Ich

3 verschiedene Programme für 3 Plattformen:

Die Quelltexte müssen für verschiedene Plattformen unabhängig entwickelt u. compiliert werden.

Nachteil: großer Aufwand.

Vorteil: Sie laufen direkt auf der Maschine u. sind damit schnell.

Dieses Vorgehen ist bei meisten üblichen Programmiersprachen im Einsatz, wie C oder Pascal.

Wie könnte man ein Programm bzw. eine Programmiersprache plattformunabhängig machen?

Vergleich: Gesprochene Sprachen:

Sie wollen sich mit drei Personen unterschiedlicher Muttersprachen unterhalten.

Die direkte Möglichkeit würde erfordern,

deren Sprachen zu lernen oder ihnen zuzumuten,

Ihre Sprache zu lernen.

Das wäre erheblicher Aufwand:

Chinesin:  
Mandarin

Grieche:  
Althebräisch

Afrikanerin:  
Kisuaheli

Ich  
Deutsch

Wie können wir eine Kommunikation mit geringerem Aufwand erreichen?

Wir stellen fest: Alle haben wir Grundkenntnisse in Englisch.

Über diese Zwischensprache kann eine Kommunikation erfolgen:

Zwischensprache:  
Englisch

Fremdsprachenkenntnisse

Fremdsprachenkenntnisse

Fremdsprachenkenntnisse

Fremdsprachenkenntnisse

Chinesin:  
Mandarin

Grieche:  
Althebräisch

Afrikanerin:  
Kisuaheli

Ich  
Deutsch

Vorteil: Kommunikation mit deutlich geringerem Aufwand

Nachteile:

* Eingeschränktes Vokabular
* Kommunikation träger

Übertragen auf Computersprachen:

Kann auf diesem Wege einer "Zwischensprache" eine Plattformunabhängigkeit erreicht werden dahingehend,

dass wir nur ein Programm schreiben müssen

u. dieses (direkt oder indirekt) auf jeder Plattform lauffähig ist?

Ja! Java ist plattformunabhängig!

Die Zwischensprache heißt: "Bytecode"

Der Java-Compiler übersetzt Java in diesen Bytecode als ersten Schritt.

Dieser Bytecode kann nicht direkt auf einer reellen Maschine ausgeführt werden,

sondern nur indirekt mit Hilfe einer Laufzeitumgebung, einer "virtuellen Maschine",

entsprechend den Fremdsprachenkenntnissen des Kommunikationspartners.

Die VM (Virtuelle Maschine) ist ein Stück Software,

dass es der konkreten Maschine erlaubt,

die maschinenunabhängige Zwischensprache auf der konkreten Maschine auszuführen.

Zwischensprache:  
Bytecode

Java-Compiler

VM

VM

Virtuelle Maschine: VM

PC:  
Windows

Mac

Unix-Station

Ich  
Java

Vorteil: Plattformunabhängigkeit

Nachteile:

1. VM muss zur Verfügung gestellt werden
2. Ausführung langsamer

Interpretation:

Sind die beiden Nachteile für den praktischen Einsatz von Relevanz?

zu 1.: VM für alle gängigen Maschinen vorhanden

zu 2.: I.d.R. nicht relevant,  
 weil Faktor Mensch eine erheblich größere Verzögerung verursacht  
 als der Umweg über die virtuelle Maschine.

Java mit der Plattformunabhängigkeit ist eine verbreitete u. bewährte Programmiersprache- bzw. Umgebung.